

Beyond the Min-Cut Bound: Deterministic Network Coding for Asynchronous Multirate Broadcast

Amy Fu, Parastoo Sadeghi and Muriel Médard

Abstract

In a single hop broadcast packet erasure network, we demonstrate that it is possible to provide multirate packet delivery outside of what is given by the network min-cut. This is achieved by using a deterministic non-block-based network coding scheme, which allows us to sidestep some of the limitations put in place by the block coding model used to determine the network capacity.

Under the network coding scheme we outline, the sender is able to transmit network coded packets above the channel rate of some receivers, while ensuring that they still experience nonzero delivery rates. Interestingly, in this generalised form of asynchronous network coded broadcast, receivers are not required to obtain knowledge of all packets transmitted so far. Instead, causal feedback from the receivers about packet erasures is used by the sender to determine a network coded transmission that will allow at least one, but often multiple receivers, to deliver their next needed packet.

Although the analysis of deterministic coding schemes is generally a difficult problem, by making some approximations we are able to obtain tractable estimates of the receivers' delivery rates, which are shown to match reasonably well with simulation. Using these estimates, we design a fairness algorithm that allocates the sender's resources so all receivers will experience fair delivery rate performance.

I. INTRODUCTION

It is well established that the information capacity of a multicast network is given by its min-cut [1], a result that has been extended to broadcast packet erasure channels in [2], [3]. This is a definitive result that cannot be violated under the system model that is considered. An important feature of this model is the requirement of block coding: that is, it is assumed that a block code is used and the original information cannot be recovered until the entire block has been received. Although block coding is practical in most situations, it is not the only possible method of encoding. As a result, it is possible that the limitations imposed by block coding do not necessarily apply in general.

A related issue is the simultaneity of decoding by the receivers in the network. Block-based codes, including rateless codes (for example, those developed for degraded broadcast channels and erasure channels [4]), impose the condition that all receivers decode all information before moving onto the next block.¹ While block-based decoding is predominantly assumed or studied, there have been some results on intermediate decoding of packets [5]–[7], albeit without considering the logical packet ordering for delivery to the application.

Linear network coding, shown in [8] to be capable of achieving network capacity, also provides some interesting counterexamples [9], [10] to the necessity of block coding restrictions. Although a large number of *block-based* network coding schemes exist [11]–[23] which enforce the complete reception of each block, there are also some non-block-based *deterministic* network coding schemes [9], [10], [24]–[28]. By relying on feedback from the receivers to determine the coded transmissions, and similar to [5]–[7], these schemes can in some cases [9], [10], [27] allow receivers to decode packets of information *prior* to receiving all information transmitted by the sender.

We consider the potential implications of using deterministic network coding in the case of a broadcast packet erasure network, comprising of a single sender connected to multiple receivers via a heterogeneous single-hop broadcast packet erasure channel. From [2], [29], the capacity of this network under the block coding requirement is given by the channel rate of the most challenged receiver, with the highest packet erasure probability, which is the min-cut of the network. As a consequence, even a single poorly-performing receiver can severely degrade the rate at which packets can be delivered in order to the application across the entire network. So a fundamental question that has not been considered before is:

What will happen to the delivery rates and simultaneity of decoding if the sender's transmission rate exceeds the channel rate of one or more receivers?

Under the traditional block coding approach, the delivery rate of these receivers would be zero. Using deterministic network coding, however, it may be possible for the receivers to continue decoding new information, despite never being able to fully receive all information that has so far been transmitted by the sender. As a result, it may be possible to maintain nonzero (in-order) packet delivery rates to all receivers – even if the sender's transmission rate exceeds their channel capacity. We demonstrate the principles behind this somewhat counterintuitive statement using the following example, which is based on the two receiver transmission scheme first presented in [24].

A. Fu and P. Sadeghi are with the Research School of Engineering, College of Engineering and Computer Science, The Australian National University, Canberra, Australia. Emails: {amy.fu;parastoo.sadeghi}@anu.edu.au.

M. Médard is with the Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. Email: medard@mit.edu.

¹The only difference is that in rateless codes the goal is generally reducing completion time rather than maintaining achievable rates.

t	$s(t)$		Rx ₁		Rx ₂
0			p_1-p_8		p_1-p_3, p_5
1	p_9	✗	p_1-p_8	✓	p_1-p_3, p_5, p_9
2	$p_9 + p_4$	✓	p_1-p_9	✓	p_1-p_5, p_9

TABLE I

AN EXAMPLE OF THE TRANSMISSION SCHEME FOR TWO RECEIVERS, SHOWING WHICH PACKETS HAVE BEEN DECODED BY EACH RECEIVER. THE NOTATION p_1-p_8 INDICATES PACKETS p_1, \dots, p_8 HAVE BEEN DECODED. A ✓ MEANS THE TRANSMITTED PACKET $s(t)$ WAS RECEIVED; A ✗ INDICATES A PACKET ERASURE.

A. Deterministic coding example

Imagine a simple broadcast network consisting of a single sender aiming to deliver a set of packets in the correct order to two receivers: a primary receiver Rx₁ with channel rate $\mu_1 < 1$ and a secondary receiver Rx₂ with channel rate $0 < \mu_2 < \mu_1$. At each time slot t the sender transmits a single packet, to which the receivers respond with an acknowledgement if they were able to receive it correctly. The sender uses this feedback to determine which coded combination should be transmitted next, as outlined in Algorithm 1. A simple example of this algorithm in progress is given in Table I. The sender's priority is to transmit packets required by Rx₁, who has delivered the most packets so far, but it will also aid Rx₂ wherever possible.

Packet transmissions intended for Rx₁ are not immediately useful to Rx₂. However Rx₂ has the ability to store and make use of any of these packets it overhears. For example, in Table I at $t = 1$, Rx₂ stores p_9 , which was intended for Rx₁. If, owing to an erasure at Rx₁, the sender retransmits p_9 , it can take the opportunity to code a second packet, p_4 , intended for Rx₂, into $s(t)$. This results in the time $t = 2$ transmission of $s(t) = p_9 + p_4$.

The inclusion of p_4 into $s(t)$ does not interfere with Rx₁'s ability to receive p_9 , since Rx₁ has already decoded all packets that could potentially be required by Rx₂. Conversely, Rx₂ will be able to decode p_4 from the transmission, since it already received p_9 in the previous time slot. When receivers' contributions to the transmission $s(t)$ do not impact each others' ability to decode, something we call a *knowledge differential transmission* occurs. These opportunities will be discussed in a more general setting later on. The end result in this example is that the sender will, as a first priority, deliver packets to Rx₁, resulting in a delivery rate of μ_1 to that receiver. However there are also occasional opportunities to deliver packets to Rx₂, resulting in a small nonzero delivery rate at Rx₂. We shall show that this idea can be generalised to R receivers, using the deterministic coding scheme studied in [9], [27].

In summary, the key idea in this paper is to break away from the block-based and rate limited schemes that are prevalent in the literature, and apply a *feedback-based, deterministic* network coding scheme as shown above. In doing so, we can achieve rates that exceed the limitations on capacity determined under the complete information acquisition requirement [2]. In a sense, this encoding scheme relaxes the notion of broadcasting common information. Although it still ensures that all receivers can decode all packets, receivers are permitted to do so at greatly differing times. By taking advantage of the leading receiver's progress and the stored packets in the buffers of lagging receivers, we can establish a more general form of broadcast, which we can call *asynchronous broadcast*, which allows receivers to deliver the same information at different rates.

Algorithm 1 Two-receiver transmission algorithm of [24].

- 1: Let $N_i(t)$ be the first undecoded packet of Rx _{i} .
 - 2: **if** Rx₂ has already received $N_1(t)$ **then**
 - 3: transmit the XOR combination $N_1(t) \oplus N_2(t)$.
 - 4: **else**
 - 5: transmit $N_1(t)$.
 - 6: **end if**
-

B. Contributions and related work

In this paper we will demonstrate that it is possible for a wireless broadcast transmission scheme to achieve multi-rate packet delivery of an ordered set of information to a heterogeneous set of receivers. By using a deterministic, non-block based transmission scheme, we are able to improve upon the capacity region established in [2].

Multi-rate broadcast to heterogeneous receivers has been previously considered in the context of applications such as video broadcast and layered streaming [30], [31], where different subsets of the information are made available to different receivers. To our knowledge, the transmission scheme we will present is the first to enable the multi-rate delivery of the same set of ordered information to all receivers.

Our transmission scheme is based on a network coding scheme first proposed in [9], which aimed to minimise the in-order packet delivery delay of a homogeneous network. By making some minor modifications to this scheme, we show that it can be

adapted for transmission in a heterogeneous broadcast network. Using an analytical model to approximate the system behaviour, we demonstrate that the delivery rate of this transmission scheme can be accurately estimated. The analysis of feedback-based deterministic coding schemes is generally a challenging problem, owing to the number of cross-receiver dependencies that must be considered. We believe that our analysis is the first attempt at this level of detail.

When implementing multi-rate broadcast, one must also consider how resources are allocated amongst receivers. We therefore introduce a measure of *fairness* whereby a transmission scheme is considered fair if the receivers' in-order packet delivery rates are proportional to their channel rate. To enable the sender to provide fair multi-rate delivery to the receivers, we introduce the concept of time sharing among different *modes*. Using a fairness algorithm to adjust the amount of time spent in each mode, we illustrate how the sender's resources can be allocated in our transmission scheme to allow good delivery rates to be achieved by all receivers under the aforementioned fairness constraint.

II. MODEL

A single sender aims to transmit a set of backlogged packets $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots$ in the correct order to a set of R receivers, $\text{Rx}_1, \dots, \text{Rx}_R$.² Time is slotted, and the sender transmits at a rate of one packet per time slot. The channel between the sender and receivers is a wireless broadcast packet erasure channel, where erasures are assumed to occur according to an ergodic Bernoulli random process. More specifically, in each time slot, each receiver Rx_i experiences an independent packet erasure with a probability μ_i , which is called its *channel rate*, as illustrated in Fig. 1, so that on average μ_i of the broadcasted packets are successfully received. Similar erasure channel models have been considered in [2], [9], [10]. We assume that the receivers' channel rates are unique, and order the receivers so that $\mu_1 > \mu_2 > \dots > \mu_R$, so that Rx_1 is the *fastest* receiver with the highest channel rate, followed by Rx_2 , Rx_3 and so on. For simplicity of notation, the set of channel rates is represented by the vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_R]$. The *receiver index* i of a receiver Rx_i is therefore a numerical ranking of its channel rate, compared to all other receivers in the system.

At each time slot t the sender broadcasts a single network coded packet $s(t)$, along with the vector of coding coefficients used. The network coding coefficients are chosen from the Galois field \mathbb{F}_M , where the field size $M \geq R$, typically the smallest power of 2 satisfying this requirement. For simplicity, we treat each packet \mathbf{p}_n as a single symbol from \mathbb{F}_M .³

Each receiver stores the packets it receives, along with the corresponding coding coefficients, in its own buffer.⁴ To decode the original packets, receivers use the information from the coding coefficients to apply Gaussian elimination [33], [34] operations, which, once enough information has been received, allow them recover the original uncoded packets. After each packet transmission, the receiver sends an acknowledgement if the packet was successfully received, or a negative acknowledgement if the packet was discarded due to an erasure. For the purposes of this work we assume the sender detects these acknowledgements without delay or error. The sender uses this information to record which packets each receiver has stored in its buffer. Based on this information, a *transmission scheme* is used to determine the packet combinations the sender will transmit. The components of the transmission schemes we will study are now outlined.

A. Transmission scheme

Here we give an overview of the transmission scheme we will be studying in this paper. As shown in Fig. 1 the transmission scheme consists of three components:

- A *packet backlog* which contains all packets not yet *delivered* (to be defined shortly) to all receivers,
- A *mode selector* which determines in which *mode* m (to be defined later) the coding scheme transmits,
- A *coding scheme* which determines what network coded combination of the packets in the sender's packet backlog will be selected for transmission.

The aim of the transmission scheme is to deliver the packets in order, though not necessarily at the same rate, to all of the receivers.

DEFINITION 1: A *packet*, \mathbf{p}_n , corresponds to the n^{th} packet in the sender's backlog, and is represented by a symbol in \mathbb{F}_M .

DEFINITION 2: A *transmission* $s(t)$ is a symbol in \mathbb{F}_M that represents the network coded packet transmitted by the sender at time t ,

$$s(t) = \sum_i \alpha_i(t) \mathbf{p}_i, \quad (1)$$

where the coding coefficients $\alpha_1(t), \alpha_2, \dots$ are symbols chosen from the field \mathbb{F}_M .

DEFINITION 3: A packet \mathbf{p}_n is *decoded* by a receiver Rx_i if Gaussian elimination on all transmissions $s(t)$ it has received so far reveals the value of \mathbf{p}_n .

²For our analysis, we assume there is an infinite number of packets to be delivered.

³In practice, a packet \mathbf{p}_n would consist of a vector of k such symbols, with $\mathbf{p}_n \in \mathbb{F}_M^k$, as presented in work such as [17], [25], [32], with the same coding operations applied to each element of the vector.

⁴It should be noted that under our transmission scheme, it is necessary for all receivers to have enough space to store all packets being coded by the sender.

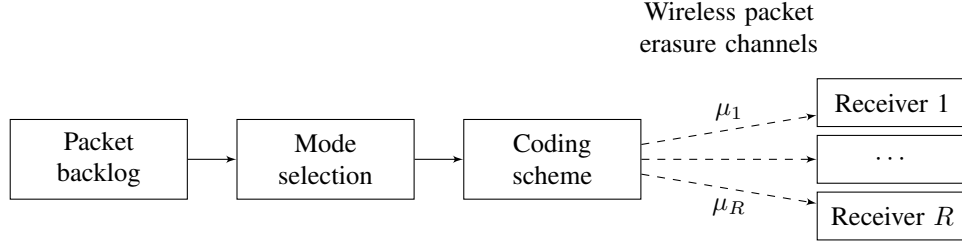


Fig. 1. A block diagram of the components of the transmission scheme.

DEFINITION 4: A packet \mathbf{p}_n is *older* than another packet \mathbf{p}_m if its packet index $n < m$, and *newer* if its packet index n is higher.

The packets in the sender's backlog are ordered from oldest to newest, so packets are delivered in order from oldest to newest.

DEFINITION 5: A packet \mathbf{p}_n is *delivered* to a receiver Rx_i if all older packets $\mathbf{p}_1, \dots, \mathbf{p}_n$ have been decoded by that receiver. $d_i(t)$ is the number of packets delivered to Rx_i by time t , so the most recently delivered packet is $\mathbf{p}_{d_i(t)}$.

DEFINITION 6: The *delivery rate* for Rx_i is given by $R_i = d_i(t)/t$, the average rate at which packets are delivered to the receiver, as $t \rightarrow \infty$.

Example: In Table I at $t = 0$, packets $\mathbf{p}_1, \dots, \mathbf{p}_8$ have been delivered to Rx_1 , while $\mathbf{p}_1, \dots, \mathbf{p}_3$ have been delivered to Rx_2 .

DEFINITION 7: At any time t , the *next needed* packet $N_i(t)$ of a receiver Rx_i is the oldest packet which has not been delivered to that receiver's buffer. From Definition 5, $N_i(t) = \mathbf{p}_{d_i(t)+1}$.

Example: In Table II, at time $t = 0$ the next needed packets of each receiver are $N_1(0) = \mathbf{p}_{11}$, $N_2(0) = \mathbf{p}_5$ and $N_3(0) = \mathbf{p}_3$.

B. Mode selection

DEFINITION 8: In *mode* m , a transmission is restricted to coding packets required by receivers $\text{Rx}_m, \dots, \text{Rx}_R$. These receivers are said to be the *mode* m receivers. A transmission made while the sender is in mode m is called a *mode* m transmission.

DEFINITION 9: The *mode vector* $\beta = [\beta_1, \dots, \beta_R]$ determines the proportion of time spent in each mode. β_m is the proportion of time spent in mode m , and $\sum_{m=1}^R \beta_m = 1$.

At each time slot, the sender randomly chooses a mode to transmit in, selecting transmission mode m with probability β_m independently of previous transmission mode choices.

C. Coding scheme

The coding scheme we will study is an adaptation of the coding scheme proposed in [9].⁵ The method for determining each transmission $s(t)$ is given in Algorithm 2.⁶ Under this coding scheme, the sender is restricted to coding from the next needed packets of the mode m receivers, so that

$$s(t) = \sum_{i=m}^R \alpha_i(t) N_i(t). \quad (2)$$

Under Algorithm 2, the sender makes a list containing each oldest undelivered packet \mathbf{p}_j and the corresponding set of receivers G_j . Then, beginning with the newest packet in the list, it adds each packet \mathbf{p}_j into $s(t)$ only if the receiver(s) in G_j would not otherwise receive an innovative packet.

In Algorithm 2, lines 13-15 ensure that a packet \mathbf{p}_j is only coded into $s(t)$ if $0 \in v_j$. From lines 7-8, this is contingent on at least one of the receivers in G_j being able to cancel out the existing packets in $s(t)$ using Gaussian elimination. Therefore Algorithm 2 has the following property.

Property 1. *In mode* m , only the next needed packets $N_m(t), \dots, N_R(t)$ of receivers $\text{Rx}_m, \dots, \text{Rx}_R$ can be coded into the transmission $s(t)$. Of these, a packet $N_i(t)$ will only be coded into $s(t)$ if it allows Rx_i to deliver its next needed packet.

⁵There are two main differences with [9]. Firstly, in this paper, rather than determining packet encodings according to a Bernoulli arrival pattern as illustrated in [9], we use modes to determine which receivers' packets should be coded. Secondly, in [9] it is suggested that any field element may be chosen, but in our implementation, we choose the smallest coefficients that will allow innovative transmissions.

⁶It can be shown that the *veto list* concept used in [9] only impacts the selection of the current coefficient, therefore Algorithm 2 presents a slightly simpler but equivalent form of the coding scheme in [9].

Example: An example of Algorithm 2 is given in Table II for the case where each group G_j consists of a single receiver. Consider $t = 2$. The coding scheme starts by setting $s(2) = \mathbf{p}_{11}$. However, Rx_2 can already cancel out this transmission using \mathbf{p}_{11} in its buffer ($0 \in v_2$ in line 13 of Algorithm 2). Hence, $s(2) = \mathbf{p}_{11} + \mathbf{p}_5$, because $N_2(2) = \mathbf{p}_5$. This is already innovative for Rx_3 . Therefore, \mathbf{p}_3 need not be added.

In [9] it is shown that, using this coding scheme, we can always find a transmission that is innovative for all receivers, if the field size $M \geq R$. It should be noted however that while every packet is innovative, receivers will still experience different delivery rates, since the packets they receive are not always contiguous, and, from Definition 5, delivery can only take place if all older packets have been decoded at that receiver.

Algorithm 2 Coding scheme.

- 1: Organise receivers $\text{Rx}_m, \dots, \text{Rx}_R$ into groups G_j , so that G_j contains all receivers Rx_i whose next needed packet $N_i(t) = \mathbf{p}_j$. Let G be the set of all groups G_j which contain at least one receiver.
 - 2: Initialise $s(t) = 0$.
 - 3: **for** each group $G_j \in G$, from high to low j , **do**
 - 4: Initialise the empty *veto list* $v_j = \{\}$.
 - 5: **for** each receiver $\text{Rx}_i \in G_j$ **do**
 - 6: Calculate r_i , the result of performing Gaussian elimination on $s(t)$ with the transmissions stored in Rx_i 's buffer.
 - 7: **if** $r_i = 0$ **then**
 - 8: $v_j \leftarrow v_j \cup \{0\}$.
 - 9: **else if** $r_i = \alpha \mathbf{p}_j$ for some field element α **then**
 - 10: $v_j \leftarrow v_j \cup \{\alpha\}$.
 - 11: **end if**
 - 12: **end for**
 - 13: **if** $0 \in v_j$ **then**
 - 14: $a_j \triangleq \min(\mathbb{F}_M \setminus v_j)$.
 - 15: Set $s(t) = s(t) + a_j \mathbf{p}_j$.
 - 16: **end if**
 - 17: **end for**
-

III. ANALYTICAL MODEL

It is difficult to model the coding scheme presented in Section II precisely. There is a complicated set of interdependencies between the packets transmitted by the sender and information stored in the receivers' buffers. The sender's transmissions are determined by the contents of the receivers' buffers, which is in turn determined by a combination of past transmissions, the receivers' erasure patterns and the effects of Gaussian elimination. These interdependencies makes this system difficult to model precisely. Therefore, in order to obtain a tractable model, we make some simplifying approximations in our delivery rate analysis. Here we list further approximations we will use in this paper to obtain a tractable model for calculating the receivers' delivery rates under our transmission scheme. In Sections V to VII we will demonstrate that the numerical results obtained under this analytical model match well with simulations.

A. Simplified encoding scheme

We now analyse Algorithm 2 and show that the majority of the time the sender implements a simplified version of the coding scheme, given in Algorithm 3. Where the conditions required to use this simplified coding scheme are satisfied, we can prove that a field size $M = 2$ is sufficient to guarantee that a sender transmission $s(t)$ which is innovative for all mode m receivers can always be found.

Approximation 1. *For the transmission schemes studied in this paper, if the receivers' channel rates are distinct and ordered so that $\mu_1 > \mu_2 > \dots > \mu_R$, then their delivery rates will also obey the ordering $R_1 > R_2 > \dots > R_R$.*

Intuitively it makes sense that higher channel rates will result in higher packet delivery rates for the corresponding receivers. Where Approximation 1 holds, the total number of packets delivered by each of the receivers will diverge over time. Therefore, after a sufficient amount of time has passed the next needed packets $N_1(t), \dots, N_R(t)$ of each receiver will also be distinct with high probability.⁷ Therefore we state the next approximation that we will use for this paper,

Approximation 2. *The next needed packet of each receiver is distinct and ordered by receiver index, so that $N_1(t) > N_2(t) > \dots > N_R(t)$.*

⁷Note however that this is not always the case: for example at $t = 0$, $N_1(t) = \dots = N_R(t) = \mathbf{p}_1$ since none of the receivers has received any transmissions yet.

Under Approximation 2, each receiver's next needed packet is distinct. When this is the case, there is only a single receiver in group G_j corresponding to each next needed packet p_j and Algorithm 2 reduces to the simplified encoding scheme presented in Algorithm 3. This simplified encoding scheme, which is based on the much smaller field size \mathbb{F}_2 , will form the basis of our analysis.

Algorithm 3 Simplified encoding scheme.

```

1: Initialise  $s(t) = 0$ .
2: for  $i = 1, \dots, R$  do
3:   Calculate  $r_i$ , the result of performing Gaussian elimination on  $s(t)$  using the packets in receiver  $Rx_i$ 's buffer.
4:   if  $r_i = 0$  then
5:      $s(t) = s(t) \oplus N_i(t)$ .
6:   end if
7: end for

```

Property 2. Under Algorithm 3, a mode m transmission is innovative to every mode m receiver, and non-innovative to receivers Rx_1, \dots, Rx_{m-1} .

Proof: Under Approximation 2, Rx_1 will have the newest next needed packet, followed by Rx_2 , Rx_3 , and so on. At each iteration i of the **for** loop, it is ensured that $s(t)$ is innovative to Rx_i . By Definition 7 and Approximation 2, $N_{i+1}(t), \dots, N_R(t)$ have already been decoded by Rx_i , so the addition of these packets to $s(t)$ in later iterations of the for loop does not affect the innovativeness of $s(t)$ to Rx_i . Algorithm 3 guarantees a transmission $s(t)$ will be found which is innovative to all receivers Rx_m, \dots, Rx_R .

If the ordering of next needed packets is maintained, packets coded for Rx_m, \dots, Rx_R are already seen by Rx_1, \dots, Rx_{m-1} . Therefore the m^{th} mode will not contain innovative information for receivers Rx_1 to Rx_{m-1} . ■

The simplified transmission scheme of Algorithm 3 also has the following property:

Property 3. Implementing Algorithm 3 with the binary field \mathbb{F}_2 is sufficient to ensure that Property 2 holds.

Proof: Under Approximation 2, the next needed packets of each receiver are distinct. Therefore each group $G_j \in G$ contains exactly one receiver. It follows that in each iteration of the **for** loop corresponding to a group $G_j \in G$, only one element can be added to the corresponding veto list v_j . Since the binary field \mathbb{F}_2 contains two elements, this field is sufficient to ensure an appropriate coefficient can always be found. ■

An example of the simplified transmission scheme for three receivers in mode 1 is given in Table II. Note that each packet coded into $s(t)$ corresponds to a receiver being able to decode its next needed packet. Two important features of this coding scheme can be observed:

- 1) It is possible for packets to be delivered to any of the receivers. For example, at time $t = 2$, Rx_2 delivers its next needed packet p_5 , and at $t = 4$ receivers Rx_1, Rx_2, Rx_3 all deliver their next needed packets p_{11}, p_7 and p_3 , respectively.
- 2) Every next needed packet coded into a transmission $s(t)$ corresponds to a receiver who can decode and hence deliver its next needed packet, if they receive the transmission. For example, at time $t = 4$, all three receivers' next needed packets are encoded. Therefore, receiving the transmission allows all three receivers to deliver in the same time slot.

t	$s(t)$	Rx_1	Rx_2	Rx_3
0		p_1-p_{10}	p_1-p_4, p_6	p_1-p_2, p_7
1	p_{11}	✗ p_1-p_{10}	✓ p_1-p_4, p_6, p_{11}	✗ p_1-p_2, p_7
2	$p_{11} + p_5$	✗ p_1-p_{10}	✓ p_1-p_6, p_{11}	✓ $p_1-p_2, p_7, p_{11} + p_5$
3	$p_{11} + p_7$	✗ p_1-p_{10}	✗ p_1-p_6, p_{11}	✓ $p_1-p_2, p_5, p_7, p_{11}$
4	$p_{11} + p_7 + p_3$	✓ p_1-p_{11}	✓ p_1-p_7, p_{11}	✓ p_1-p_3, p_7, p_{11}

TABLE II

AN EXAMPLE OF MODE 1 TRANSMISSION, AND THE EFFECT ON RECEIVERS' BUFFERS. TICKS AND CROSSES REPRESENT RECEIVERS' CHANNEL STATES (SUCCESSFUL AND UNSUCCESSFUL RECEPTION, RESPECTIVELY) AT EACH TRANSMISSION.

B. Buffer model

We outline the model we will use in our analysis of the receivers' buffer's contents. The packets stored in the buffer serve a dual purpose. Firstly, they provide information that will facilitate the decoding of the received packets. Secondly, they store information about packets that will need to be delivered at some point in the future.

For the purpose of analysis, we can imagine that every receiver's buffer has space allocated for all of the packets in the sender's backlog. This can be imagined as a large vector, with each element corresponding to a packet, ordered left to right from oldest to newest. The associated coding coefficients are also stored along with each element of the vector, so that the receiver has the information required to perform Gaussian elimination. As illustrated in Fig. 2, each position n in the buffer corresponds to the packet \mathbf{p}_n . The following will be used to describe the contents of a receiver's buffer.

DEFINITION 10: For a receiver Rx_i , The *buffer coefficient matrix* \mathbf{B}_i stores the coding coefficients associated with each buffer position. Each packet \mathbf{p}_n corresponds to the n^{th} column of the buffer coefficient matrix, which we denote by $\mathbf{B}_i(n)$. If packet \mathbf{p}_i is coded into the combination stored at \mathbf{p}_n , then the i^{th} element of that column vector will correspond to its coding coefficient. Otherwise, this element will be 0. If buffer position n is empty, then $\mathbf{B}_i(n) = \mathbf{0}$.

To decode the packets stored in the buffer, identical Gaussian elimination operations are applied to the columns of the buffer coefficient matrix, and the packets in the buffer. Once a column of the buffer coefficient matrix $\mathbf{B}_i(n)$ has been reduced to a standard basis vector e_n , then the corresponding packet has been decoded.

DEFINITION 11: For a receiver Rx_i , the *buffer fill vector* \mathbf{b}_i indicates whether any information has been stored at each position n of the buffer. We say that $\mathbf{b}_i(n) = 0$ if $\mathbf{B}_i(n)$ is the zero vector, otherwise $\mathbf{b}_i(n) = 1$.

Generally, the receiver will be required to store both coded and uncoded packets. Algorithm 4 outlines procedure for storing a received transmission $s(t)$ in the buffer. This procedure can be divided into three stages: using Gaussian elimination to reduce $s(t)$, storing the result in the correct buffer position, and using this new packet to further reduce other packets in the buffer. We now give an example of Algorithm 4 at work.

Example: In Table II, at time $t = 2$, Rx_3 cannot further reduce $s(2) = \mathbf{p}_{11} + \mathbf{p}_5$, nor can $s(2)$ further reduce any of the packets stored by Rx_3 . Therefore the received packet $\mathbf{p}_{11} + \mathbf{p}_5$ will be stored at buffer position 11. By contrast, at time $t = 3$ the presence of packets \mathbf{p}_7 and $\mathbf{p}_{11} + \mathbf{p}_5$ allows the received packet $s(3) = \mathbf{p}_{11} + \mathbf{p}_7$ to be reduced down to $s'(3) = \mathbf{p}_5$. After storing $s'(3) = \mathbf{p}_5$ at buffer position 5, we use it to reduce $\mathbf{p}_{11} + \mathbf{p}_5$ down to \mathbf{p}_{11} which will replace the content of buffer position 11.

Algorithm 4 Adding a received packet to the buffer of Rx_i .

- 1: Find $s'(t)$, the remainder when Gaussian elimination is applied to $s(t)$ using the columns of \mathbf{B}_i .
 - 2: **if** $s'(t) \neq \mathbf{0}$ **then**
 - 3: Add $s'(t)$ to buffer position n , where n is the highest packet index in $s'(t)$.
 - 4: Use $s'(t)$ to apply Gaussian elimination to the non-empty buffer positions, to obtain column echelon form for \mathbf{B}_i .
 - 5: **end if**
-

Property 4. For every receiver Rx_i , Algorithm 4 ensures that for every filled element n in the buffer, \mathbf{p}_n is the packet with the highest index coded.

Proof: Algorithm 4 only modifies the contents of the buffer at lines 3 and 4. At line 3, $s'(t)$ is stored at the correct position. Since Gaussian elimination has already been applied to $s'(t)$, applying line 4 will not change the highest packet index of any of the stored combinations. ■

We now introduce some terminology that will be useful later.

DEFINITION 12: A packet \mathbf{p}_n is *seen* to receiver Rx_i if $\mathbf{b}_i(n) = 1$, i.e. there is a packet stored at buffer position n .⁸

Example: In Table II, at time $t = 2$, receiver Rx_3 has packets $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_7, \mathbf{p}_{11} + \mathbf{p}_5$ in its buffer. In our model of the buffer, positions 1, 2, 7 and 11 are filled, therefore $\mathbf{b}_3(1) = \mathbf{b}_3(2) = \mathbf{b}_3(7) = \mathbf{b}_3(11) = 1$ and packets $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_7$ and \mathbf{p}_{11} have been seen. The nonzero columns of the buffer coefficient matrix are given by $\mathbf{B}_3(1) = e_1$, $\mathbf{B}_3(2) = e_2$, $\mathbf{B}_3(7) = e_7$, and $\mathbf{B}_3(11) = e_{11} + e_5$.

DEFINITION 13: A transmission $s(t)$ is *innovative* to a receiver Rx_i if $s(t)$ cannot be computed from packets in Rx_i 's buffer.

This method of representing the buffer has the following useful property:

Property 5. A receiver Rx_i 's next needed packet is always the first unseen packet in its buffer. In other words, its next needed packet is given by $N_i(t) = \mathbf{p}_n$, where n is the first buffer position for which $\mathbf{b}_i(n) = 0$.

Proof: If $\mathbf{b}_i(1) = \dots = \mathbf{b}_i(n-1) = 1$, then packets $\mathbf{p}_1, \dots, \mathbf{p}_{n-1}$ must have been decoded and delivered to the receiver (since we have received $n-1$ innovative combinations of $n-1$ packets). By Property 4, if \mathbf{p}_n has been delivered by Rx_i , then we must have $\mathbf{b}_i(n) = 1$. Since $\mathbf{b}_i(n) = 0$, it follows that the first unseen packet \mathbf{p}_n is the next needed packet of Rx_i . ■

1) *Regions of the buffer:* As illustrated in Fig. 2, our model of the buffer of a receiver Rx_i can be broken down into several distinct regions. The *delivered region* consists of all packets $\mathbf{p}_1, \dots, \mathbf{p}_{d_i(t)}$ which have been delivered to the receiver.

⁸In this paper we use the definition of *seen* introduced in [9]. Note that this differs from the definition used in [10], [25].

By definition this is immediately followed by $N_i(t) = p_{d_i(t)+1}$, the oldest undelivered packet in the buffer. Following this is the *packet density region*, consisting of combinations of transmitted but undelivered packets $p_{d_i(t)+2}, \dots, N_1(t)$. In this region it is possible for Rx_i to overhear both coded and uncoded transmissions intended for faster receivers. All packets lying beyond $N_1(t)$ form the *empty region* where no packets are yet stored, since no information about them has been broadcasted yet.

For a receiver Rx_i , there are three possible outcomes regarding the storage of every received packet in the buffer.

- If $s(t)$ is not innovative to Rx_i , then the packet will be discarded.
- If $s(t)$ is innovative and allows $N_i(t)$ to be decoded, then $s'(t)$ will be stored at position $d_i(t) + 1$. This results in the delivery of one or more new packets, so that $d_i(t+1) > d_i(t)$.
- If $s(t)$ is innovative but it does not help decode $N_i(t)$, then the transmission is stored in the packet density region of the buffer according to Algorithm 4.

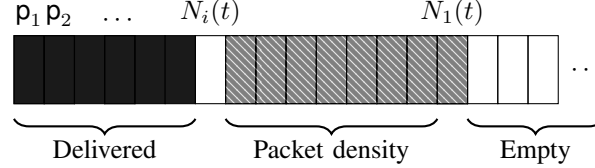


Fig. 2. Our model of receiver Rx_i 's buffer. Dark coloured packets have been all received and delivered; some fraction of the shaded packets have been received; white packets have not been received. Packets are ordered left to right by ascending packet index.

2) *Subregions*: Taking a closer look at the packet density region of Fig. 2, this region can be further divided into subregions $1, 2, \dots, R$, each with its own distinct packet density. This is illustrated in Fig. 3. The reason each of these regions exists, is that the transmissions from each mode m contribute to the packet density of each subregion. However, it is not a uniform contribution. Mode 1 transmissions evenly cover packets in subregions $1, 2, \dots, R$, while mode 2 transmissions cover subregions $2, \dots, R$, and so on. The impact of these transmissions depends on how frequently they occur, which is in part determined by the mode vector β . For a given receiver Rx_r , the primary subregion of interest is subregion r , which gives us the packet density in the subregion where Rx_r is delivering its next needed packets. The packet density of subregion r is called the *buffer density* B_r , which will be useful in determining the delivery rate later on.

DEFINITION 14: For a receiver Rx_r , the *buffer density* B_r is defined as the average proportion of buffer positions filled inside subregion r ,

$$B_r = \begin{cases} 0, & r = 1 \\ \frac{\sum_{j=d_r(t)+2}^{d_{r-1}(t)} b_r(j)}{(d_{r-1}(t) - d_r(t) - 1)}, & 1 < r \leq R. \end{cases} \quad (3)$$

This is illustrated in Fig. 3.

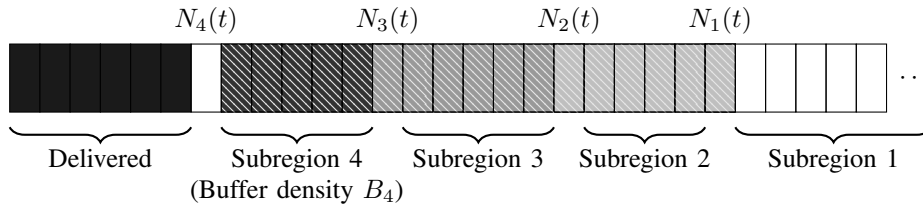


Fig. 3. Subregions within the packet density region of receiver Rx_4 's buffer. Darker colouring corresponds to a higher packet density; white indicates a packet has not been received. Packets are ordered left to right by ascending index.

3) *Buffer model approximations*: In our delivery rate analysis of the transmission scheme, we model our system for the simple coding scheme presented in Section III-A. However, even to this coding scheme we apply some further simplification for tractability. Therefore we now present a list of approximations that will be used to model the receivers' buffers.⁹

Approximation 3. All packets stored in a receiver Rx_i 's buffer are uncoded, so if $b_i(n) = 1$, then $B_i(n) = e_n$. This means that once a receiver Rx_i has seen a packet p_n , we assume decoding of it as well.

This approximation results a somewhat optimistic estimate of the probability of decoding. However, this is not an unreasonable approximation as our coding scheme is *near-systematic*, meaning that whenever Property 2 permits, the sender transmits an

⁹It can be observed that all of the approximations in this section are used to remove the more unpredictable elements of cross-receiver interactions. For Rx_1 and Rx_2 , the effective lack of interaction with other receivers means that when analysing these two receivers, our approximations will be completely accurate. As the receiver index increases however, the accuracy of these approximations will slowly decrease.

μ	β	Percentage of undecoded packets
$[0.8, 0.6, 0.4, 0.2]$	$[0.85, 0.05, 0.05, 0.05]$	0.0%, 0.0%, 2.66%, 10.26%
$[0.9, 0.8, \dots, 0.2]$	$[0.65, 0.05, \dots, 0.05]$	0.0%, 0.01%, 0.63%, 1.06%, 2.88%, 5.23%, 11.5%16.22%
$[0.8, 0.75, \dots, 0.45]$	$[1, 0, \dots, 0]$	0.0%, 0.03%, 0.31%, 0.60%, 3.20%, 6.10%, 11.93%, 12.07%

TABLE III
PERCENTAGE OF UNDECODED PACKETS IN EACH RECEIVER'S BUFFER, UNDER THE INDICATED VALUES OF μ AND β .

uncoded packet. And of the coded transmissions received, many will be reduced to uncoded packets when they are stored in the buffer, through Gaussian elimination. From our simulations we observe that, after Gaussian elimination has been applied, the vast majority of packets stored in the buffer are uncoded. Some examples are given in Table III. In general, we observe that a few factors influence the accuracy of Approximation 3. Higher channel rates and a lower receiver index i will reduce the fraction of undecoded packets a receiver Rx_i has stored in its buffer. On the other hand, more time spent in modes $2, \dots, R$ will reduce the proportion of undecoded packets stored in the receivers' buffers.

Example: In Table II at time $t = 1$, we see a typical situation where the sender sends an uncoded packet, in this case $s(t) = p_{11}$. Although occasionally a coded packet may be stored in the receiver's buffer, for example the buffer of Rx_3 at $t = 2$, often Gaussian elimination will help decode the packet, illustrated for Rx_3 at $t = 3$.

Approximation 3 leads to the following property, which simplifies our model of the receivers' packet delivery mechanisms. This property will be useful for the knowledge differential calculations of Section VI.

Property 6. *For a receiver Rx_r to decode its next needed packet $N_r(t)$, every packet coded into $s(t)$ apart from $N_r(t)$ must have been seen by Rx_r .*

DEFINITION 15: The *leader* in a mode m transmission is the receiver Rx_m .

DEFINITION 16: A *leader transmission* of a packet p_n is a transmission $s(t)$ where p_n is the packet with the highest index coded into the transmission, i.e. the next needed packet of the leader. If the sender is in mode m when this transmission occurs, we can more precisely describe it as a *mode m leader transmission*.

Example: If the transmission $s(t) = p_3 + p_5 + p_{10}$, then it is a leader transmission of packet p_{10} , the newest packet in the transmission. If the sender was in mode 2 at time t , then $s(t)$ is a mode 2 leader transmission of p_{10} .

We can further simplify our model of the receivers' packet delivery behaviour by limiting the ways in which packets can be seen.

Approximation 4. *A receiver Rx_i can only have seen an undelivered packet p_n through the reception of a leader transmission of p_n .*

In practice, due to the effects of Gaussian elimination, it is possible for a receiver Rx_i to see an undelivered packet in the packet density region at any time slot. But for the purpose of analysis, we only allow the receiver to see a packet by the most likely method, namely receiving a leader transmission. As a result of this approximation, we obtain the following property, which will also be useful in Section VI.

Property 7. *For a packet p_n in a receiver's packet density region to be seen, Rx_r must have received a leader transmission of p_n .*

One issue this property raises is that, since there is a common sender, the same number of leader transmissions of each packet will be sent to the receivers. As a result, we can expect some correlation between which packets each receiver has seen. These cross receiver dependencies would significantly increase the complexity of our analysis, so for the sake of simplicity we make the following approximation.

Approximation 5. *The number of mode m leader transmissions of a packet p_n is independent of the number of mode b leader transmissions of p_n , for all $m \neq b$.*

It can be observed that most of the approximations made in this section are designed to minimise the number of cross-receiver interactions we will need to take into account in our calculations. With the exception of Approximation 2, our analytical model is exact for receivers Rx_1 and Rx_2 . But, as the receiver index increases, so does the impact of cross receiver interactions on the receiver's performance. As a result, it is expected that our analytical model will lose some accuracy as the number of receivers increases. Our simulation results indicate that the delivery rates predicted by the analytical model are accurate for $R = 4$ and 8 , however the computational complexity of the simulations prevent us from investigating higher values of R with accuracy.

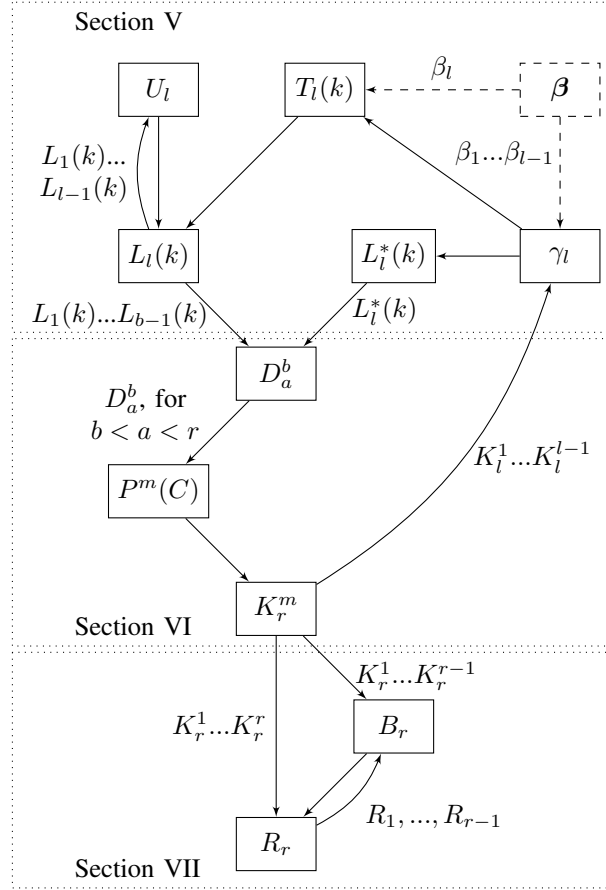


Fig. 4. Dependency graph for quantities calculated in Sections V to VII. Labeled arrows indicate which values are required to calculate the next block, while unlabeled arrows indicate the following block depends directly on the value from the previous block. Curved arrows indicate dependencies on previously calculated values.

IV. OUTLINE OF DELIVERY RATE ANALYSIS

In this section we outline the steps that will be taken in Sections V to VII to determine the delivery rate R_r for each receiver R_{x_r} under the transmission scheme outlined in Section II-C. The estimation of the delivery rate is a challenging task, given the somewhat complicated dependency between sender transmissions and the contents of each of the receivers' buffers. The approximations made in Section III go some way toward simplifying these relationships, but the process of calculating the receivers' delivery rates still requires some detail. A map of the quantities we will calculate on our journey towards finding the delivery rate is illustrated in Fig. 4, with definitions given in Table IV for quick reference. The values in this figure will be discussed later, but for now we simply observe that a number of dependencies exist between different components in the calculation.

To divide the problem into manageable pieces, we make the observation that there are two ways in which a packet may be delivered to a receiver. Either:

- The packet was previously seen from leader transmissions of the packet in other modes,
- The packet is delivered through a *knowledge differential*, while it is the receiver's next needed packet.

The rate analysis is therefore divided into three parts. Analysis of leader transmissions takes place in Section V, while knowledge differentials are studied in Section VI. The results from these two sections are combined to determine the delivery rate in Section VII. Each of these sections, corresponding to the dotted rectangles in Fig. 4, will now be briefly discussed.

A. Leader transmissions

In Section V, we characterise the number of mode l leader transmissions of a packet \mathbf{p}_n in the packet backlog. The process for calculating this probability is illustrated by the top box of Fig. 4. By Approximation 4, this characterisation is important because it can be used to determine the likelihood that a packet \mathbf{p}_n has been seen by a receiver.

There are three possible cases to consider when finding the likelihood that a packet \mathbf{p}_n has been transmitted k times in mode l . These depend on R_{x_l} 's progress towards delivering \mathbf{p}_n .

U_l	Probability that a packet will not be seen from Rx_l from mode $1, \dots, l-1$ transmissions.
$T_l(k)$	Probability that there will be k total mode l leader transmissions of a packet, given that it was not seen from mode $1, \dots, l-1$ leader transmissions.
β	The mode vector set by the sender (see Definition 9).
$L_l(k)$	Probability that a total of k mode l leader transmissions were made of a packet that has been delivered by Rx_l .
$L_l^*(k)$	Probability that so far, there have been exactly k mode l leader transmissions of $N_l(t)$, Rx_l 's current next needed packet.
γ_l	Probability of a knowledge differential transmission for Rx_l .
D_a^b	Probability that the next needed packet $N_b(t)$ of a receiver Rx_b is seen by another receiver Rx_a , where $a > b$.
$P^m(C)$	Probability of a coding C being transmitted in mode m .
K_r^m	Probability that Rx_r has a knowledge differential while the sender is in mode m .
B_r	Expected buffer density.
R_r	Estimated delivery rate for receiver Rx_r .

TABLE IV
DEFINITION OF THE VALUES USED IN FIG. 4.

- If $n \leq d_l(t)$ then p_n has already been delivered by Rx_l , and $L_l(k)$, calculated in Section V-D, gives the probability there were k mode l leader transmissions of p_n .
- If $n = d_l(t) + 1$, then $p_n = N_l(t)$, and k takes the probability distribution $L_l^*(k)$, calculated in Section V-C.
- If $n > d_l(t) + 1$, then there have been no mode l leader transmissions of p_n so far, i.e. $k = 0$, so we do not need to consider the possibility of any mode l leader transmissions of these packets.

The probabilities $L_l(k)$ and $L_l^*(k)$ are calculated in Section V.

B. Knowledge differentials

In Section VI we find the probability of a knowledge differential. This process is illustrated by the middle box in Fig. 4. Knowledge differentials are a key concept in this paper, as they are necessary to allow receivers to deliver the remaining packets that were not seen from leader transmissions.

DEFINITION 17: At time t , a receiver Rx_r has a *knowledge differential* if the portion of $s(t)$ coded for faster receivers Rx_1, \dots, Rx_{r-1} can be decoded by Rx_r using the information stored in its buffer. If the sender is in mode m , then the probability of a knowledge differential is given by K_r^m .

If $s(t)$ meets this criterion, it is a *knowledge differential transmission*. Otherwise, it is a *non-knowledge differential transmission*.

Knowledge differentials occur when portions of the transmission $s(t)$ intended for faster receivers do not affect Rx_r 's ability to decode the transmission. Knowledge differentials make it possible to provide different delivery rates to each of the receivers, since some packets will be received effectively without interference from the faster receivers. Whenever a knowledge differential transmission is received, Rx_r is able to decode and hence deliver its next needed packet $N_r(t)$.

Property 8. A receiver Rx_i 's next needed packet $N_i(t)$ can only be seen through the reception of a knowledge differential transmission.

Combining Property 1 and Approximation 2, a new packet can be delivered to Rx_r iff its next needed packet $N_r(t)$ is encoded into the transmission $s(t)$. In these time slots, the sender can take advantage of the knowledge differential to effectively send Rx_r an uncoded packet.

There are two components required to calculate K_r^m : D_a^b , the probability that a receiver Rx_a has a Rx_b knowledge difference, and $P^m(C)$, the probability of a particular coding C . These are now discussed.

1) *Knowledge differences*: From Property 6, each of the packets coded into $s(t)$ for other receivers must be seen by a receiver Rx_a in order to obtain a knowledge differential. The probability of each of these packets being seen is given by the probability of a Rx_b knowledge difference, defined as follows.

DEFINITION 18: At time t , a receiver Rx_a obtains a Rx_b knowledge difference if $\mathbf{b}_a(d_b(t) + 1) = 1$, i.e. Rx_a has seen $N_b(t)$. The probability of Rx_a having a Rx_b knowledge difference is D_a^b . If $a \leq b$, then we set $D_a^b = 1$.¹⁰

This means that when Rx_a has a Rx_b knowledge difference, even if $N_b(t)$ is coded in the sender transmission $s(t)$, it will not affect Rx_a 's ability to decode $s(t)$. Examples of this can be seen in Table II. At times $t = 2, 3, 4$, Rx_2 has seen Rx_1 's next needed packet, \mathbf{p}_{11} . At time $t = 4$, Rx_3 has seen both Rx_1 and Rx_2 's next needed packets, \mathbf{p}_{11} and \mathbf{p}_7 , respectively.

It is important to note that knowledge differences are a common occurrence, since the broadcast nature of the system means that all transmissions may be overheard by any of the receivers. A packet seen by one receiver, but not another, causes a knowledge difference to occur in the future. Generally the higher a receiver's channel rate, the better its chances of obtaining a knowledge difference, since they are more likely to overhear packets that are missed by other receivers.

2) *Codings*: The probability of a knowledge differential K_r^m depends on which of the receivers' next needed packets are coded into $s(t)$. Therefore in order to determine the probability of a knowledge differential, it is first necessary to find the probability of every possible coding, C .

DEFINITION 19: The *transmission coding* C is an R -length binary vector representing which of the receivers' next needed packets have been coded into $s(t)$. If the r^{th} element of C , $C(r) = 1$, then $N_r(t)$ is one of the packets coded into $s(t)$. Otherwise, $C(r) = 0$ and $s(t)$ does not code $N_r(t)$.

Example: If $R = 4$ and $s(t) = N_1(t) + N_3(t) + N_4(t)$, then $C = [1, 0, 1, 1]$. If $R = 5$ and $s(t) = N_2(t) + N_4(t)$, then $C = [0, 1, 0, 1, 0]$.

C. Delivery rate

In Section VII we will find the delivery rate R_r . This process is illustrated in the bottom box in Fig. 4. The knowledge differential probabilities established in Section VI are used to determine the delivery rate of all of the receivers as follows.

By Property 2, every mode l transmission received by a receiver Rx_r is innovative for $r \geq l$. Therefore, it either allows the delivery of $N_r(t)$, or is stored in the packet density region. Knowledge differential transmissions allow delivery of next needed packets, and non-knowledge differential transmissions contribute seen packets to the buffer. The buffer density B_r (recall Definition 14) can be determined from the probability of non-knowledge differential transmissions and the delivery rates of receivers Rx_1, \dots, Rx_{r-1} .

For a receiver Rx_r , the buffer density B_r gives us the fraction of packets which are already seen by Rx_r , prior to the contribution of knowledge differential transmissions. The delivery rate can be found by determining how quickly knowledge differential transmissions can fill the remaining fraction of unseen packets.

V. LEADER TRANSMISSIONS

In this section, we find the number of mode l leader transmissions of any packet \mathbf{p}_n we can expect, at any time t . All calculations in this section are implicitly dependent on the mode vector β , however for readability we omit direct mention of β in our notation.

A. Mode 1 and 2 leader transmissions

We begin with a walkthrough of the mode 1 and 2 leader transmissions of an arbitrary packet \mathbf{p}_n . Recall the definitions of a seen packet (Definition 12) and a leader transmission (Definition 16).

The first time a packet \mathbf{p}_n will be transmitted by the sender is as a mode 1 leader transmission. This will only happen once $\mathbf{p}_n = N_1(t)$, i.e. it is the next needed packet of Rx_1 . The total number of times k that \mathbf{p}_n is transmitted in mode 1 is simply the number of mode 1 transmissions it takes until it is seen by Rx_1 .

Example: Say that at time $t = 10$, the receiver Rx_1 has delivered $\mathbf{p}_1, \dots, \mathbf{p}_7$, so its next needed packet is $N_1(10) = \mathbf{p}_8$. There are mode 1 transmissions at $t = 10, 12, 13$ and the corresponding Rx_1 channel state at these time slots is $\mathbf{x}, \mathbf{x}, \checkmark$. In this case, at times $t = 10, 11, 12$, there will have been $k = 1, 2, 3$ mode 1 leader transmissions of \mathbf{p}_8 respectively, after which it is seen by Rx_1 . There are no more mode 1 leader transmissions of \mathbf{p}_8 once it is seen. After this time \mathbf{p}_9 will become Rx_1 's next needed packet, and there will have been a total of $k = 3$ mode 1 leader transmissions of \mathbf{p}_8 .

Similarly for mode 2, once $N_2(t) = \mathbf{p}_n$, all mode 2 transmissions will be leader transmissions of \mathbf{p}_n until it is seen by Rx_2 . The difference from the mode 1 case is that some of the earlier mode 1 leader transmissions of a packet \mathbf{p}_n may have been

¹⁰We set $D_a^b = 1$ when $a \leq b$, since if $a < b$ then Rx_b has already decoded $N_a(t)$, and when $a = b$, a receiver cannot affect its own probability of decoding.

$\mathbf{p}_n =$	\mathbf{p}_1	\dots	$N_3(t)$	\dots	$N_2(t)$	\dots	$N_1(t)$
$m = 1$	$L_1(k)$						$L_1^*(k)$
$m = 2$	$L_2(k)$				$L_2^*(k)$		
$m = 3$	$L_3(k)$	$L_3^*(k)$					

TABLE V

A SCHEMATIC OF THE THREE-RECEIVER PROBABILITY THAT THE SENDER WILL TRANSMIT k MODE l LEADER TRANSMISSIONS OF EACH PACKET \mathbf{p}_n .

overheard by Rx₂, therefore there is a probability that \mathbf{p}_n is seen *prior* to any mode 2 transmissions. If this is the case, then $k = 0$ since no mode 2 leader transmissions of \mathbf{p}_n are required.

Example: At time $t = 11$, the receiver Rx₂ has delivered $\mathbf{p}_1, \dots, \mathbf{p}_4$ and also has seen \mathbf{p}_6 and \mathbf{p}_8 . Mode 2 transmissions occur at $t = 11, 14, 17$, and the corresponding Rx₂ channel state at these time slots is $\checkmark, \times, \checkmark$. At $t = 11$, $N_2(11) = \mathbf{p}_5$, and Rx₂ receives the first mode 2 leader transmission of \mathbf{p}_5 . As a result, there is exactly one mode 2 leader transmission of \mathbf{p}_5 . Since \mathbf{p}_6 is already seen by Rx₂, it is immediately delivered and there are no mode 2 leader transmissions of that packet. At time $t = 14$, the sender skips straight to $N_2(14) = \mathbf{p}_7$. Rx₂ experiences an erasure at this time slot, so \mathbf{p}_7 is transmitted again at $t = 17$, giving $k = 2$ total mode 2 leader transmissions of \mathbf{p}_7 . Since \mathbf{p}_8 is already seen, the sender will skip to \mathbf{p}_9 as Rx₂'s next needed packet the next time a mode 2 leader transmission occurs.

From these examples we can make the following observation about mode l leader transmissions.

Property 9. *There can only be mode l leader transmissions of a packet \mathbf{p}_n while it is unseen by Rx _{l} .*

Proof: By Property 1, only the receivers' next needed packets can be coded into a transmission. From Property 5, the next needed packet of a receiver must always be unseen by that receiver. It follows that there will be no mode l leader transmissions of a packet \mathbf{p}_n that is seen by Rx _{l} . ■

B. Buffer position and seen packets

We now examine the distribution of k , the number of mode l leader transmissions, in terms of buffer position. At a given time t , a receiver Rx _{l} has delivered packets $\mathbf{p}_1, \dots, \mathbf{p}_{d_l(t)}$. This means that there have also been mode l leader transmissions of some or all of the packets in the corresponding region of the buffer, as illustrated in Table V.

We define the probability $L_l(k)$ of there being k mode l leader transmissions of a given packet \mathbf{p}_n for $n \leq d_l(t)$. By Approximation 5 this probability distribution is independent for each value of l , so the number of leader transmissions k of a packet \mathbf{p}_n in mode l is not correlated to the number of leader transmissions made in other modes.

In contrast, the next needed packet $N_l(t) = \mathbf{p}_{d_l(t)+1}$ has a slightly different probability distribution. Since it is in the process of being delivered, there may or may not have been previous mode l leader transmissions of this packet. Therefore, we allow for this packet to have a potentially different probability of k mode l leader transmissions and denote it by $L_l^*(k)$, the probability that there have been exactly k mode l leader transmissions of the packet so far.

1) *Seen packets:* From this information, we can make some general comments about the methods by which a next needed packet $N_b(t)$ of a receiver Rx _{b} can be seen by another receiver Rx _{a} , based on its position n in the receiver's buffer. We are primarily interested in studying the next needed packets, since only these packets will be encoded by the sender.

If $b > a$ then Rx _{a} has already seen $N_b(t)$, since by Approximation 2 it has already been delivered to Rx _{a} . If $a > b$, then by Approximation 4, the probability of a packet $N_b(t)$ being seen by another receiver Rx _{a} depends on the number of leader transmissions of the packet in modes $1, \dots, b$.¹¹ Generally, the more leader transmissions of $N_b(t)$ there have been, the more likely it is that Rx _{a} has already seen this packet.

C. Calculating $L_l^*(k)$

We begin with the simpler case: finding the probability $L_l^*(k)$ that $N_l(t)$, the next needed packet of Rx _{l} , has been transmitted k times so far in mode l .

The sender only transmits mode l leader transmissions of a new packet once the previous next needed packet has been seen by Rx _{l} . Recall from Approximation 4 that a next needed packet can be received by two methods:

- Mode l leader transmissions¹²
- Mode $1, \dots, l - 1$ knowledge differential transmissions.

¹¹Modes $b + 1, \dots, R$ do not contribute leader transmissions because $N_b(t)$ has not yet been transmitted in those modes.

¹²Mode l transmissions are also knowledge differential transmissions, but because we are counting the number of mode l leader transmissions, we treat them separately.

The probability of a mode l transmission is simply β_l , while the probability of a mode $i = 1, \dots, l-1$ knowledge differential transmission for Rx_l is given by

$$\gamma_l = \sum_{m=1}^{l-1} \beta_m K_l^m, \quad (4)$$

where K_l^i , the probability that Rx_l has a knowledge differential from a mode i transmission, will be calculated in Section VI.

So from Rx_l 's perspective, there are four possible actions from the sender at each time slot, which we now list. The sender can transmit:

- 1) A mode l transmission, with probability β_l
- 2) A mode $1, \dots, l-1$ knowledge differential transmission, with probability γ_l
- 3) A mode $1, \dots, l-1$ non-knowledge differential transmission
- 4) A mode $l+1, \dots, R$ transmission

The probability of each outcome can be multiplied by μ_l or $\bar{\mu}_l$ to give the probability that it will or will not be received by Rx_l , respectively.

Note that once the sender begins mode l leader transmissions of a new packet, it will continue to do so until it is seen, i.e. Rx_l receives either a mode l or mode $1, \dots, l-1$ knowledge differential transmission of that packet. Therefore, $L_l^*(k)$ can be considered as the probability there have been k unreceived mode l transmissions since the most recently received mode l or knowledge differential transmission.

In this context, mode $1, \dots, l-1$ non-knowledge differential transmissions as well as mode $l+1, \dots, R$ transmissions are not of interest, since they have no impact on the number of mode l leader transmissions. These correspond to items 3 and 4 in our list of sender actions. We can ignore these transmissions by normalising the probability of each event of interest by $\beta_l + \mu_l \gamma_l$, the likelihood that item 1 or 2 from our list will occur.

Then the probability that an unreceived mode l transmission occurs before either a knowledge differential transmission or a received mode l transmission is given by

$$\frac{\bar{\mu}_l \beta_l}{\beta_l + \mu_l \gamma_l} \quad (5)$$

and the probability of a received mode l or knowledge differential transmission occurring instead of an unreceived mode l transmission is given by

$$\frac{\mu_l \beta_l + \mu_l \gamma_l}{\beta_l + \mu_l \gamma_l}. \quad (6)$$

We are now ready to determine the value of $L_l^*(k)$. For there to have been exactly k prior mode l leader transmissions of Rx_l 's current next packet $N_l(t)$, two requirements must be satisfied. Firstly, for there have been at least k transmissions of $N_l(t)$, the previous k transmissions of interest must have been unreceived mode l transmissions. Secondly, to ensure that there have been no more than k transmissions of $N_l(t)$, these unreceived mode l transmissions must have been preceded by either a received mode l or knowledge differential transmission. Combining (5) and (6), $L_l^*(k)$ is then given by

$$L_l^*(k) = \left(\frac{\bar{\mu}_l \beta_l}{\beta_l + \mu_l \gamma_l} \right)^k \frac{\mu_l \beta_l + \mu_l \gamma_l}{\beta_l + \mu_l \gamma_l}. \quad (7)$$

D. Calculating $L_l(k)$

In this part, we calculate the probability $L_l(k)$ that a packet \mathbf{p}_n that has been delivered by Rx_l was transmitted k times in total in mode l . The calculation of $L_l(k)$ depends on two separate probabilities: U_l , the probability that \mathbf{p}_n was unseen by Rx_l from mode $1, \dots, l-1$ leader transmissions of \mathbf{p}_n , and $T_l(k)$, the probability that, given \mathbf{p}_n was unseen Rx_l from mode $1, \dots, l-1$ transmissions, there were exactly k mode l leader transmissions of \mathbf{p}_n . We now calculate these values.

1) *Calculating U_l* : We begin by calculating U_l , the probability that a packet \mathbf{p}_n will remain unseen to receiver Rx_l from mode $1, \dots, l-1$ leader transmissions of the packet. The probability that Rx_l will not see \mathbf{p}_n from mode i transmissions, where $i < l$, is given by

$$\sum_{k=0}^{\infty} L_i(k) \bar{\mu}_l^k. \quad (8)$$

Under Approximation 5, we can assume that seen packets are distributed uniformly across the buffer. Therefore the probability that a packet \mathbf{p}_n will still be unseen by Rx_l from mode $1, \dots, l-1$ transmissions is given by

$$U_l = \prod_{i=1}^{l-1} \sum_{k=0}^{\infty} L_i(k) \bar{\mu}_l^k. \quad (9)$$

2) *Calculating $T_l(k)$* : Next we find $T_l(k)$, the probability that there will be exactly k mode l leader transmissions of a packet \mathbf{p}_n , given it was unseen from mode $1, \dots, l-1$ leader transmissions. As in Section V-C, only mode l transmissions and received knowledge differential transmissions impact the value of k . So again, we ignore all other transmissions by normalising the probability of each relevant transmission by $\beta_l + \gamma_l \mu_l$.

If $k = 0$, then a knowledge differential transmission must have been received before any mode l leader transmissions of \mathbf{p}_n could occur. This occurs with probability

$$T_l(0) = \frac{\mu_l \gamma_l}{\beta_l + \mu_l \gamma_l}. \quad (10)$$

For $k > 0$, using (5) we find that the probability of $k-1$ unreceived mode l transmissions occurring prior to any receptions of knowledge differential or mode l transmissions is given by

$$\left(\frac{\bar{\mu}_l \beta_l}{\beta_l + \mu_l \gamma_l} \right)^{k-1}. \quad (11)$$

Following these $k-1$ unreceived mode l transmissions, one of two things can happen to limit the sender to exactly k mode l leader transmissions of \mathbf{p}_n .

- A received mode l transmission occurs,
- An unreceived mode l transmission occurs, followed by a received knowledge differential transmission.

The probability that either of two events will occur is given by

$$\frac{\mu_l \beta_l}{\beta_l + \mu_l \gamma_l} + \frac{\bar{\mu}_l \beta_l \mu_l \gamma_l}{(\beta_l + \mu_l \gamma_l)^2}. \quad (12)$$

Combining (10), (11) and (12), the probability that exactly k mode l transmissions are required to obtain a previously unseen packet $N_l(t)$ is given by

$$T_l(k) = \begin{cases} \frac{\mu_l \gamma_l}{\beta_l + \mu_l \gamma_l}, & k = 0 \\ \frac{\mu_l \beta_l (\beta_l + \gamma_l) (\bar{\mu}_l \beta_l)^{k-1}}{(\beta_l + \mu_l \gamma_l)^{k+1}}, & k > 0. \end{cases} \quad (13)$$

We now scale this by $U_l(k)$, which from (9) is the probability that \mathbf{p}_n is unseen after mode $1, \dots, l-1$ transmissions, to obtain the probability that the sender will transmit a packet k times,

$$L_l(k) = \begin{cases} 1 - U_l + U_l T_l(0), & k = 0 \\ T_l(k) U_l, & k > 0. \end{cases} \quad (14)$$

E. Comparison with simulation

We examine the accuracy of our $L_l(k)$ calculations by comparing the calculated and simulation-based probabilities in Fig. 5. We observe that calculated and simulation values match extremely well for receivers $l = 1, 2, 3$, but loses some accuracy for Rx_4 , although it does follow the same general trend. This can be attributed to the fact our approximations lead to some loss of accuracy, particularly for higher receiver indices.

We now examine the shape of $L_l(k)$, for different values of l . For $l = 1$, every packet must be transmitted at least once for Rx_1 to receive it. Therefore there is no possibility of $k = 0$ for this receiver. The most likely case is $k = 1$, since Rx_1 's high channel rate means the majority of the time, a packet will be received after one transmission. For the same reason, $L_1(k)$ decays relatively quickly as k increases. At higher values of l , however, it is reasonably likely that $k = 0$ because the receiver can overhear the packet from faster receivers' leader transmissions. Although the number of leader transmissions a receiver can overhear increases as the receiver index increases, the receiver's channel rate also decreases. In this simulation, it results in a smaller probability of $k = 0$ as l increases. We also observe that the distribution of $L_l(k)$ decays more slowly for higher values of l . This can also be attributed to the lower channel rates of the corresponding receivers.

VI. KNOWLEDGE DIFFERENTIALS

In this section, we will determine K_r^m , the probability that a receiver Rx_r obtains a knowledge differential from a mode m transmission, given the mode vector β . Through the dependencies on $L_l^*(k)$ and $L_l(k)$, all calculations in this section are also implicitly dependent on β , but for readability we omit direct mention of β in our notation.

A. Knowledge differences

We calculate D_a^b , the probability of a receiver Rx_a having a Rx_b knowledge difference. Recall from Definition 18 that if $a \leq b$ then $D_a^b = 1$.

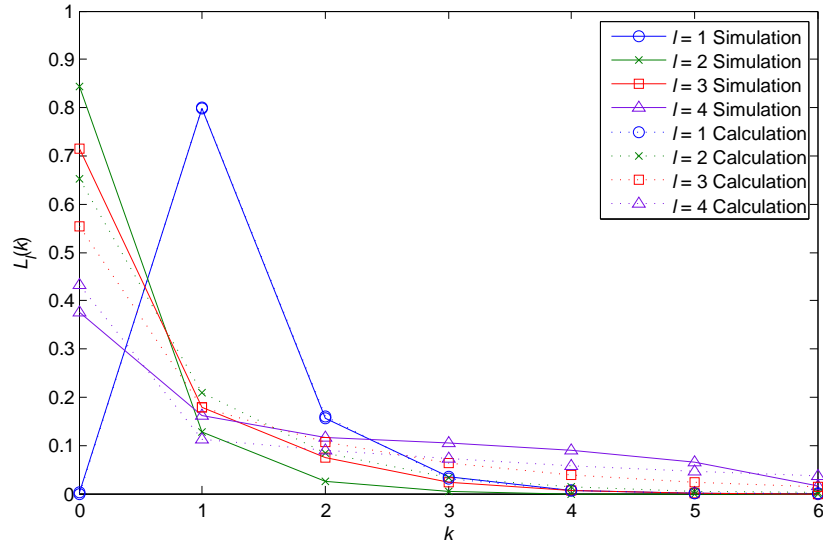


Fig. 5. Probability of k leader transmissions of a packet, for $\mu = [0.8, 0.6, 0.4, 0.2]$ and $\beta = [0.85, 0.05, 0.05, 0.05]$.

If $b < a$, then the probability D_a^b that a receiver Rx_a has a Rx_b knowledge difference is given by the probability that Rx_a has seen $N_b(t)$ from at least one of the mode $1, \dots, b$ transmissions. This can be determined from the complementary probability of the packet *not* being seen from mode $1, \dots, b$ transmissions, which can be calculated using (8) and (9).

$$D_a^b = 1 - \left(\sum_{k=0}^{\infty} L_b^*(k) \bar{\mu}_a^k \right) \prod_{l=1}^{b-1} \sum_{k=0}^{\infty} L_l(k) \bar{\mu}_a^k. \quad (15)$$

Recall that in mode b transmissions, $N_b(t)$ is the next needed packet of Rx_b and hence $L_b^*(k)$ is used instead of $L_l(k)$.

B. Probability of a coding

We next establish the probability $P^m(C)$ that in mode m , a particular coding C will occur. The notation $P^m(\cdot)$ represents the mode m probability of the event inside the brackets.

DEFINITION 20: The *subcoding* C_r takes the first r elements of the coding C . C_r is the r -length binary vector representing which of receivers Rx_1 to Rx_r 's next needed packets have been coded into $s(t)$.

Example: For the coding $C = [1, 0, 1, 1]$, the corresponding subcodings are $C_1 = [1]$, $C_2 = [1, 0]$, $C_3 = [1, 0, 1]$ and $C_4 = [1, 0, 1, 1]$.

Combining Property 1, Property 6 and Approximation 3, a packet $N_r(t)$ will only be coded into $s(t)$ if Rx_r has seen all of the other packets coded into $s(t)$. But by Approximation 2, receiver Rx_r has already decoded the next needed packets of receivers Rx_{r+1}, \dots, Rx_R . Therefore the subcoding C_r corresponds to the portion of the transmission coding C that will affect whether or not $N_r(t)$ is coded into $s(t)$, and therefore the value of $C(r)$. From this information, we can break down finding the probability $P^m(C)$ into the problem of finding the probability, $P^m(C_r)$ for $r = 1, \dots, R$.

In practice, it is usually not necessary to perform calculations starting from $P^m(C_1)$. A mode m transmission will always code the next needed packet of the leader, so

$$P^m(C(m) = 1) = 1. \quad (16)$$

Furthermore, since $N_1(t), \dots, N_{m-1}(t)$ will not be coded into a mode m transmission,

$$C(1) = \dots = C(m-1) = 0. \quad (17)$$

Combining these two pieces of information, we obtain the result,

$$P^m(C_m) = \begin{cases} 1, & \text{for } C_m = [0, \dots, 0, 1] \\ 0, & \text{for all other } C_m. \end{cases} \quad (18)$$

This means that we can start our calculations from C_m , since all possible mode m subcodings begin with $[0, \dots, 0, 1]$. Equation (18) can be used as the base case to determine the probability of the other subcodings C_r , for $r = m+1, \dots, R$.

Under Property 6 the only way for a packet $N_r(t)$ to be coded is if the corresponding receiver Rx_r has seen all packets coded into the transmission $s(t)$. Then the probability of $N_r(t)$ being added to $s(t)$ is the probability that Rx_r has a Rx_b

knowledge difference for all receivers Rx_b corresponding to packets coded into $s(t)$,

$$\begin{aligned} P^m(C(r) = 1|C_{r-1}) &= \prod_{b \in f_1^m(C_{r-1})} D_r^b \\ &= \prod_{b \in f_1^m(C)} D_r^b, \end{aligned} \quad (19)$$

where for a vector v , we define $f_1^m(v)$ as the list of all indices $r \geq m$ where $v(r) = 1$, and $f_0^m(v)$ as the list of all indices $r > m$ where $v(r) = 0$. Note that the second part of this equation is true because $D_r^b = 1$ for all $r \leq b$.

The probability $P^m(C_r)$ of each subcoding C_r can then be defined in terms of the previously calculated probability $P^m(C_{r-1})$,

$$P^m(C(r) = 1) = P^m(C_{r-1})P^m(C(r) = 1|C_{r-1}), \quad (20)$$

$$P^m(C(r) = 0) = P^m(C_{r-1})P^m(C(r) = 0|C_{r-1}). \quad (21)$$

Combining this pair of equations with (19), we can derive the probability of a particular subcoding C_r :

$$P^m(C_r) = \left(\prod_{a \in f_1^m(C_r)} \prod_{b \in f_1^m(C_r)} D_a^b \right) \left(\prod_{a \in f_0^m(C_r)} 1 - \prod_{b \in f_1^m(C_r)} D_a^b \right), \quad (22)$$

if the first m elements of C_r are $C_m = [0, \dots, 0, 1]$. Otherwise, $P^m(C_r) = 0$.

C. Knowledge differentials

We now find the probability K_r^m of a knowledge differential occurring for a receiver Rx_r under mode m transmission.

By Algorithm 3 receiver Rx_r 's next needed packet is coded iff a knowledge differential occurs. This means that we can determine the probability of a knowledge differential by finding the probability that the next needed packet of Rx_r is coded into a transmission,

$$\begin{aligned} K_r^m &= P^m(C(r) = 1) \\ &= \sum_C C(r) P^m(C) \\ &= \sum_{C_r} C_r(r) P^m(C_r). \end{aligned} \quad (23)$$

It can be noted that, in the special case where $r = m$, we can substitute (18) to obtain

$$K_m^m = 1. \quad (24)$$

Property 10. The probability K_r^m of a mode m knowledge differential for a receiver Rx_r depends only on the ratio of $\beta_1 : \beta_2 : \dots : \beta_r$.

Proof: We use induction, starting with the base case of $r = 1$. From (24) and Property 2, $K_1^1 = 1$, and $K_1^m = 0$ for $m > 1$. This is true regardless of what is happening to the other receivers, so for the fastest receiver corresponding to $r = 1$, Property 10 is satisfied.

For each subsequent value of r , we examine the dependency of K_r^m on β using Fig. 4. We observe that dependencies on the mode vector β are fed in through two variables: $T_l(k)$ and $L_l^*(k)$, via γ_l . In particular, K_r^m is calculated from $T_1(k), \dots, T_{r-1}(k)$ and $L_r^*(k)$. Therefore we need only prove that $T_l(k)$ and $L_l^*(k)$ depend on the ratio $\beta_1 : \beta_2 : \dots : \beta_l$.

From (4), if K_l^i satisfies Property 10 for $i < l$, then scaling β_1, \dots, β_l by some factor $\omega > 0$ will also scale γ_l by a factor of ω . We then make the key observation that, in each case of (13), every term in the numerator and denominator has the same multiplicity of either β_l or γ_l . This means that multiplying β_1, \dots, β_l and γ_l by a factor of ω in (13) will not change the value of $T_l(k)$. Therefore, scaling β_1, \dots, β_l by some value ω will not affect the value of $L_l(k)$.

In (7), we similarly observe that the same multiplicity of β_l and γ_l in the numerator and denominator of $L_l^*(k)$. Therefore both $T_l(k)$ and $L_l^*(k)$ depend only on the ratio $\beta_1 : \beta_2 : \dots : \beta_l$. ■

D. Simulation results

We examine the accuracy of our knowledge differential calculations. We can observe here that, for $r = 1, 2, 3$ the calculated values of K_r^m match the simulated values extremely well, while at $r = 4$ some inaccuracy observed at $m = 1, 2$. This can be attributed to the cross-receiver effects from Rx_2 and Rx_3 in modes $m = 1, 2$. For $m = 3, 4$ only Rx_3 and Rx_4 's packets are coded. Since only Rx_3 can affect Rx_4 's knowledge differential probability, there are few cross-receiver interactions to take into account when determining K_r^m , so the calculations remain accurate.

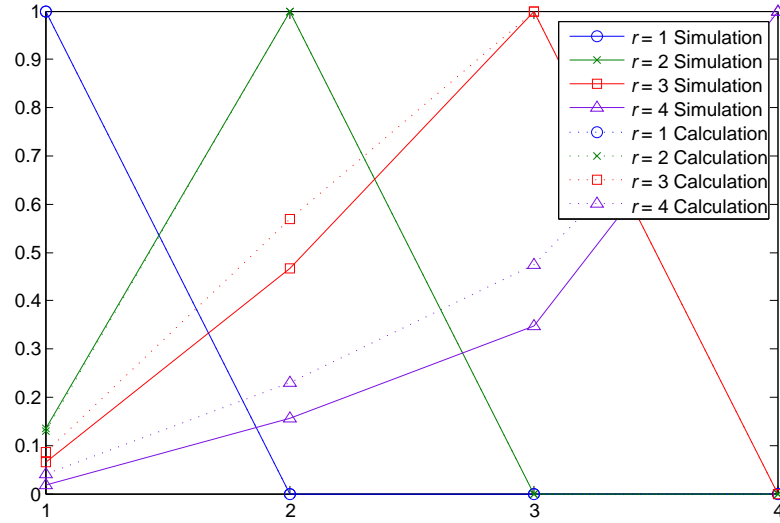


Fig. 6. Probability of a knowledge differential for a receiver Rx_r in mode m , for $\mu = [0.8, 0.6, 0.4, 0.2]$ and $\beta = [0.85, 0.05, 0.05, 0.05]$.

The shape of the K_r^m can be explained as follows. For a given receiver Rx_r , K_r^m is a relatively low probability at $m = 1$, that increases as m increases. This increasing probability can be attributed to the lesser number of coded receivers, which makes it more likely that a knowledge differential will occur. At mode $m = r$, K_r^m reaches its maximum value, with $K_r^r = 1$. This matches with (24). In modes $m > r$, transmissions are no longer innovative, in line with Property 2. As a result, the probability of a knowledge difference is $K_r^m = 0$.

VII. DELIVERY RATES

Now that the probability of K_r^m of a mode m knowledge differential is known, we are ready to estimate R_r , the delivery rate of each receiver Rx_r . It can be observed in Fig. 4 that B_r and R_r are recursively dependent on one another.

As outlined in Section III-B, we can divide all transmissions received by Rx_r into three categories. Those that are non-innovative, those that allow the delivery of $N_r(t)$, and those that contribute to the buffer density. We use this information to estimate the delivery rate for the receivers.

A. Buffer density

We firstly determine B_r , receiver Rx_r 's buffer density. The first step is to determine which of the sender's transmissions can contribute to the buffer density. From Section III-B1, a transmission must be both innovative and a non-knowledge differential transmission in order to be stored in the packet density region of the buffer. By Property 2, transmissions from modes $m = r + 1, \dots, R$ will be noninnovative to Rx_r . From Section VI, K_r^m is the fraction of the sender's mode m transmissions that will result in knowledge differentials for Rx_r (for $m \leq r$), and from (24) $K_r^r = 1$. Therefore only the $\overline{K_r^m}$ fraction of mode $m = 1, \dots, r - 1$ non-knowledge differential transmissions can contribute to Rx_r 's buffer density.

Since the distribution of leader transmissions studied in Section V is uniform, it is reasonable to say that within a subregion, the buffer density will also be uniform. Therefore, the buffer density B_r is the average number of non-knowledge differential transmissions received from modes $m = 1, \dots, r - 1$ at each packet index in subregion r .

We consider the contribution from each mode $m < r$. For a given mode m , we assume all received mode m non-knowledge differential packets are distributed uniformly between $p_1, \dots, N_m(t)$. Then the average number of sender transmissions corresponding to each packet index in subregions m, \dots, R is the inverse of Rx_m 's delivery rate, $1/R_m$. Of the β_m fraction of time spent in mode m , $\overline{K_r^m}$ of the packets transmitted will be non-knowledge differential transmissions for Rx_r . These are distributed uniformly across $p_1, \dots, N_m(t)$ so there is on average β_m/R_m mode m transmissions per packet. Of these, μ_r will be received by Rx_r . So summing over all modes $m = 1, \dots, r - 1$,

$$B_r = \mu_r \sum_{m=1}^{r-1} \frac{\beta_m \overline{K_r^m}}{R_m}. \quad (25)$$

For $r = 1$ there would be no summation, so we use this as the base case for our recursive calculation,

$$B_1 = 0. \quad (26)$$

B. Delivery rate

The task of delivering new packets to a receiver R_{x_r} is equivalently the process of making the unseen packets in the buffer seen to R_{x_r} , using knowledge differential transmissions. On average, B_r of the undelivered packets in the buffer have already been seen by the receiver. Therefore it is necessary to fill in the remaining $(1 - B_r)$ fraction of packets using knowledge differential transmissions. On average, μ_r of the knowledge differential transmissions will be received. Therefore, the delivery rate is given by

$$R_r = \frac{\mu_r \sum_{m=1}^r \beta_m K_r^m}{1 - B_r}. \quad (27)$$

Here we must sum the knowledge differentials contributed by each mode. Note that the summation only goes from 1 to r , since from Property 2, a mode m transmission can only be innovative to receivers R_{x_m}, \dots, R_{x_R} .

We can observe from (25) and (27) that there is a recursive relationship between the delivery rates of the receivers. Note that B_r only depends on R_1, \dots, R_{r-1} and K_r^m , calculated in Section VI, while R_r only depends on B_r and K_r^m . Therefore starting from the R_{x_1} base case, where the delivery rate is $R_1^m = \beta_1 \mu_1$ and the buffer density is $B_1^m = 0$, we can sequentially calculate the delivery rates of all receivers.

C. Rate calculation algorithm

The process for calculating the delivery rate has many recursive dependencies that make it difficult to determine how to implement. Therefore, we provide Algorithm 5 to outline the order in which variables shown in Fig. 4 can be calculated.

Making use of the fact that a receiver's delivery rate is only influenced by faster receivers, we sequentially calculate the delivery rate R_r for each receiver R_{x_r} starting from $r = 1, \dots, R$. Following the same structure as Sections V to VII-B, we perform the leader transmission calculations, followed by knowledge differential calculations and finally the delivery rate calculations.

It is interesting to note that, in the special case of $r = 1$, the calculations reduce significantly. Because there are no earlier receivers for R_{x_1} 's behaviour to be influenced by, the calculation dependencies illustrated by curved arrows in Fig. 4 can be removed.¹³

Algorithm 5 Delivery rate calculation.

```

1: Set  $D_a^b = 1$  for all  $a \geq b$ .
2: Set  $K_r^m = 1$  for all  $r = m$ , and  $K_r^m = 0$  for  $r < m$ .
3: for  $m = 1, \dots, R$  do
4:   Calculate  $\gamma_m$ , using  $K_m^1, \dots, K_m^{m-1}$ .
5:   Calculate  $L_m^*(k)$ , using  $\gamma_m$ .
6:   Calculate  $U_m$  using  $L_1, \dots, L_{m-1}(k)$ .
7:   Calculate  $T_m(k)$  using  $\gamma_m$ .
8:   Calculate  $L_m(k)$  using  $U_m$  and  $T_m(k)$ .
9:   for  $a = m + 1, \dots, R$  do
10:    Calculate  $D_a^m$  using  $L_1(k), \dots, L_{m-1}$  and  $L_m^*(k)$ .
11:   end for
12:   for  $n = 1, \dots, m$  do
13:    for all  $C_m$  do
14:     Calculate  $P^n(C_m)$  using  $D_a^b$  for  $b < a < m$ .
15:    end for
16:    for  $r = n, \dots, m$  do
17:     Calculate  $K_r^n$  using all  $P^n(C_r)$ .
18:    end for
19:   end for
20:   Calculate  $B_m$  using  $K_m^1, \dots, K_m^{m-1}$  and  $R_1, \dots, R_{m-1}$ .
21:   Calculate  $R_m$  using  $K_m^1, \dots, K_m^m$  and  $B_m$ .
22: end for

```

D. Simulation results

We now compare the simulated and calculated delivery rates for the receivers. Fig. 7 illustrates the simulated and calculated delivery rates of a set of receivers under the settings listed in Table VI.

¹³In our calculations, empty summations are equal to zero, and empty products are equal to one.

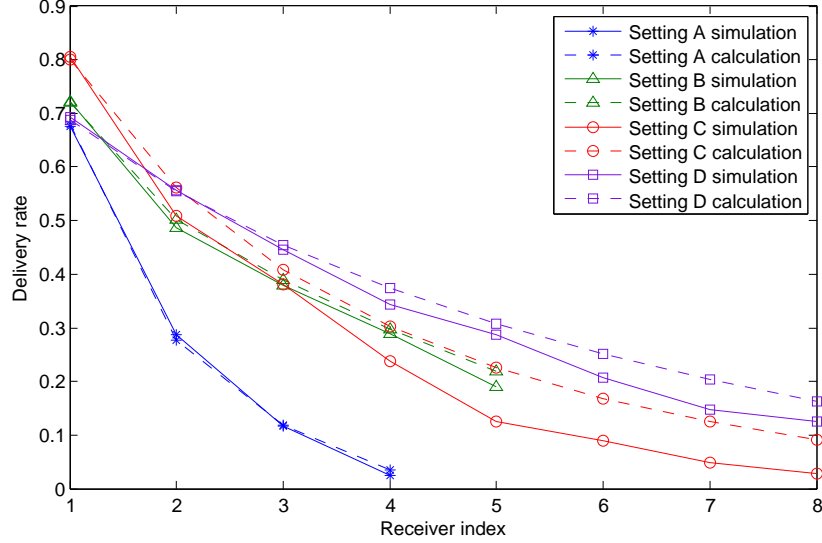


Fig. 7. Probability of a knowledge differential for a receiver Rx_r in mode m , for the simulation settings given in Table VI.

Setting	μ	β
A	[0.8, 0.6, 0.4, 0.2]	[0.85, 0.05, 0.05, 0.05]
B	[0.9, 0.8, 0.7, 0.6, 0.5]	[0.8, 0.05, \dots , 0.05]
C	[0.8, 0.75, 0.7, \dots , 0.45]	[1, 0, \dots , 0]
D	[0.8, 0.75, \dots , 0.45]	[0.86, 0.02, \dots , 0.02]

TABLE VI
SIMULATION CONDITIONS USED IN FIG. 7.

Under Settings A and B, the calculated delivery rates match well with the values observed in simulation. In settings C and D on the other hand, the approximate calculations are fairly accurate for the first few receivers, but gradually lose accuracy for higher receiver indices. The calculated values tend to be a little optimistic as the receiver index grows, since at higher receiver indices, Approximation 3 loses its accuracy. Setting C represents the extreme case, where all sender transmissions are allocated to mode 1. For a receiver Rx_r , the closer the mode m is to r , the more accurate the knowledge differential estimate will be. Therefore greater the amount of time allocated to higher modes, the more accurate the calculated rates will be.¹⁴

It is interesting to note that, for Setting A, despite some slight inaccuracies in the knowledge differential estimates of Fig. 6, the delivery rates under both simulation and calculation match quite well. In part, this can be attributed to the inverse relationship between the buffer density and the probability of knowledge differential that can be observed in (25). As a result of this relationship, inaccuracies in the calculated probability of a knowledge differential are partially mitigated by their inverse effect on the buffer density.

VIII. FAIRNESS

We now use the delivery rate analysis of Sections VI and VII to determine the value of the mode vector β that will result in fair delivery rates for all receivers. We define fairness as follows:

DEFINITION 21: A transmission scheme is *fair* for all receivers if their delivery ratios are equal, so that

$$\frac{R_1}{\mu_1} = \frac{R_2}{\mu_2} = \dots = \frac{R_R}{\mu_R}. \quad (28)$$

Our aim then is to achieve equal *delivery ratios* for all receivers, where the delivery ratio Q_r of a receiver Rx_r is defined as

$$Q_r = \frac{R_r}{\mu_r} \quad (29)$$

In practice, the method we will outline can be modified to equalise any set of delivery ratios, so long as doing so will

¹⁴Note, however, that β must be restricted to values that maintain the ordering of the receivers under Approximation 2.

maintain the condition

$$R_1 \geq R_2 \geq \dots \geq R_R. \quad (30)$$

A. The rate scaling property

Before we can begin, it is necessary to highlight the key property of our transmission scheme that will help us build our fairness algorithm.

Property 11. *Scaling β_1, \dots, β_r by some amount $\omega > 0$ does not affect the buffer density of receivers Rx_1, \dots, Rx_r , and scales their delivery rates by ω .*

Proof: We will prove this property using induction. Let us substitute $\beta'_m = \omega\beta_m$ for all $m = 1, \dots, r$ into (27) and (25). Using the notation $B_r(\beta)$ and $R_r(\beta)$ to show the explicit dependency on the mode vector β , we obtain

$$B_r(\beta') = \mu_r \sum_{m=1}^{r-1} \frac{\omega\beta_m \overline{K_r^m}}{R_m(\beta')} \quad (31)$$

$$R_r(\beta') = \frac{\sum_{m=1}^r \omega\beta_m K_r^m}{1 - B_r(\beta')} \quad (32)$$

As the base case, $R_1(\beta') = \omega\beta_1\mu_1$ and $B_1(\beta') = 0$. $R_1(\beta')$ is linearly dependent on ω , and $B_1(\beta')$ is a constant. Therefore Property 11 holds for the base case.

We now show that, if Property 11 is true for Rx_1, \dots, Rx_{r-1} , then it also holds for Rx_r . From Property 10, we know that scaling β_1, \dots, β_r by ω will not affect the probability K_r^m of a knowledge differential. Now, assume that $R_m(\beta') = \omega R_m(\beta)$ and $B_m(\beta') = B_m(\beta)$ for $m < r$. Then by substituting these values into (32) and (31), we obtain

$$\begin{aligned} B_r(\beta') &= \mu_r \sum_{m=1}^{r-1} \frac{\omega\beta_m \overline{K_r^m}}{\omega R_m(\beta)} \\ &= B_r(\beta) \end{aligned} \quad (33)$$

and

$$\begin{aligned} R_r(\beta') &= \frac{\sum_{m=1}^r \omega\beta_m K_r^m}{1 - B_r(\beta)} \\ &= \omega R_r(\beta). \end{aligned} \quad (34)$$

Therefore, Property 11 holds for all receivers. ■

This is a key result since it tells us that, for a receiver Rx_r , scaling the time spent in modes $1, \dots, r$ by a factor of ω will scale the delivery rates R_1, \dots, R_r by a factor of ω as well. This will give us a predictable way to determine the receivers' delivery rates as we adjust the value of the mode vector β .

B. Fairness algorithm

We now outline the fairness algorithm used to determine the value of β required to achieve delivery rate fairness. Recall that, in Section VII, we calculated both the expected buffer density and delivery rates, given β . The aim now is to determine the value of β that will result in fair delivery rates for all receivers.

The receivers' delivery ratios can be matched using the procedure outlined in Algorithm 6. This fairness algorithm performs two functions as it iterates through each value of r . Firstly, it scales down the amount of time allocated to modes $1, \dots, r-1$ by the calculated quantity $1 - \omega[r]$. Secondly, it allocates the remaining $\omega[r]$ fraction of time to mode r . By Properties 2 and 11, neither of these actions affects the delivery ratio of receivers $1, \dots, r-1$. Therefore, Algorithm 6 ensures that equalised delivery ratios will remain matched through subsequent iterations of the algorithm. After the r^{th} iteration is complete, $\beta[r]$ will allow the sender to achieve $Q_1[r] = \dots = Q_r[r]$.

1) *Notation:* To reflect the changing values at each iteration, we append the variables with an index $[r]$ to represent the updated values as of the r^{th} iteration. $\beta_m[r]$ represents the time allocated to mode m after iteration r , and $\omega[r]$ is an equalisation factor which determines $\beta[r]$ based on $\beta[r-1]$, the value at the previous iteration. The delivery ratio $Q_m[r]$ of each receiver Rx_r is also determined from $\beta[r]$. At each iteration r , the value $\beta[r]$ is updated to match the delivery ratio of Rx_r to that of receivers Rx_1, \dots, Rx_{r-1} . In the remainder of this section, we will explicitly show any dependencies on $\beta[r]$, for example $K_r^m(\beta[r])$ represents receiver Rx_r 's mode m knowledge differential probability given that $\beta[r]$ is used as the mode vector.

We begin by allocating all transmissions to mode 1, so that $\beta_1[1] = 1$, and $\beta_m[1] = 0$, for $m = 2, \dots, R$. By (24), all mode 1 transmissions are knowledge differential transmissions for Rx_1 . Therefore the delivery ratio for Rx_1 after iteration 1 is $Q_1[1] = 1$.

Algorithm 6 Fairness algorithm.

```

1: Set  $\beta_1[1] = 1$ , and  $\beta_2[1], \dots, \beta_R[1] = 0$ .
2: for  $r = 2$  to  $R$  do
3:   Calculate  $Q_{r-1}[r-1]$  and  $Q_r[r-1]$  using (29), substituting  $\beta = \beta[r-1]$ .
4:   Calculate  $\omega[r]$  from (37).
5:   Set  $\beta_r[r] = \omega[r]$ 
6:   for  $m = 1$  to  $r-1$  do
7:     Set  $\beta_m[r] = (1 - \omega[r])\beta_m[r-1]$ 
8:   end for
9: end for
10: Set  $\beta = \beta[r]$ .

```

TABLE VII
SETTINGS USED IN FAIRNESS ALGORITHM SIMULATIONS.

Setting	μ
A	[0.8, 0.7, 0.6, 0.5]
B	[0.9, 0.8, \dots , 0.5]
C	[0.9, 0.85, \dots , 0.55]

In each iteration $2 \leq r \leq R$ of the for loop, we find a value of $\beta[r]$ which will achieve $Q_1[r] = \dots = Q_r[r]$, equalising the delivery ratio of Rx_r with that of receivers Rx_1 to Rx_{r-1} . This is determined by reducing the time spent in modes $1, \dots, r-1$ by a factor of $(1 - \omega[r])$ to allow a fraction of time $\omega[r]$ to be spent in mode r .

By Property 11, this has the effect of scaling $Q_1[r], \dots, Q_{r-1}[r]$ by a factor $1 - \omega[r]$, so that

$$Q_m[r] = (1 - \omega[r])Q_m[r-1] \quad (35)$$

for all $m < r$.

Combining (27), (29), Property 11 and (24), the delivery ratio of receiver Rx_r in iteration r is given by

$$\begin{aligned} Q_r[r] &= \frac{\sum_{m=1}^r \beta_m[r] K_r^m(\beta[r])}{1 - B_r[r]} \\ &= (1 - \omega[r])Q_r[r-1] + \frac{\omega[r]}{1 - B_r(\beta[r-1])}. \end{aligned} \quad (36)$$

Therefore to equalise the delivery ratios of Rx_r and Rx_{r-1} the task is to find the value of $\omega[r]$ for which $Q_r[r] = Q_{r-1}[r]$. This can be solved using (35) and (36) to give

$$\omega[r] = \frac{Q_{r-1}[r-1] - Q_r[r-1]}{Q_r[r-1] - Q_{r-1}[r-1] + \frac{1}{1 - B_r(\beta[r-1])}}. \quad (37)$$

For $m = 1, \dots, r-1$ the new values of $\beta_m[r]$ are now given by

$$\beta_m[r] = \begin{cases} (1 - \omega[r])\beta_m[r-1], & m < r \\ \omega[r], & m = r \\ 0, & m > r \end{cases} \quad (38)$$

This process is repeated for each receiver to equalise the delivery ratio of each subsequent receiver, until all receivers are equal. After this process is complete, the amount of time that should be spent in each mode m to achieve fairness is given by

$$\beta = \beta[R]. \quad (39)$$

The delivery ratio of every receiver Rx_r using the mode vector β will be equal. Substituting (24) and (26) into (27), the delivery rate of Rx_1 is found to be $\mu_1\beta_1$, so by (29), the delivery ratio of Rx_1 is $Q_1 = \beta_1[R]$. Therefore, using the mode vector β will give the delivery ratios

$$Q_1 = Q_2 = \dots = Q_R = \beta_1[R]. \quad (40)$$

C. Simulation results

We can now compare the delivery rates achieved both in theory and in practice. The channel rates for each setting we will investigate is given in Table VII.

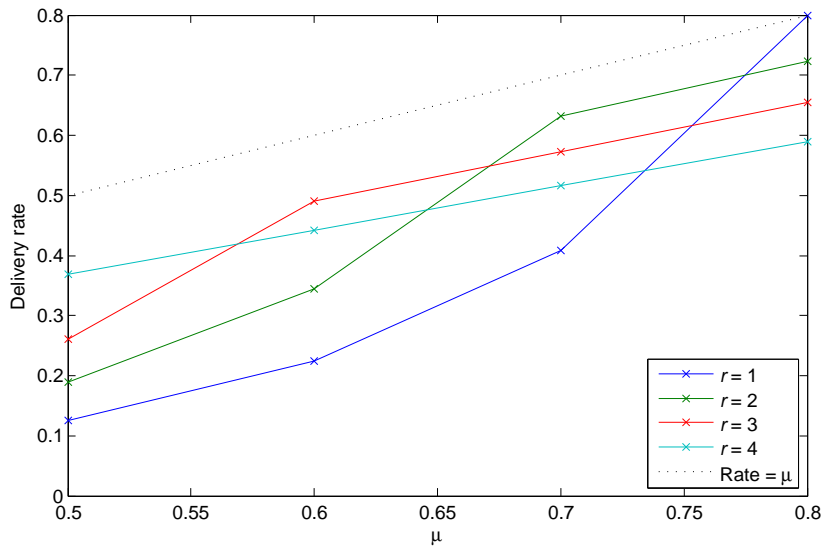


Fig. 8. Calculated delivery rates achieved after each iteration r of Algorithm 6, under Setting A from Table VII. These are compared against μ , the channel rate for each receiver.

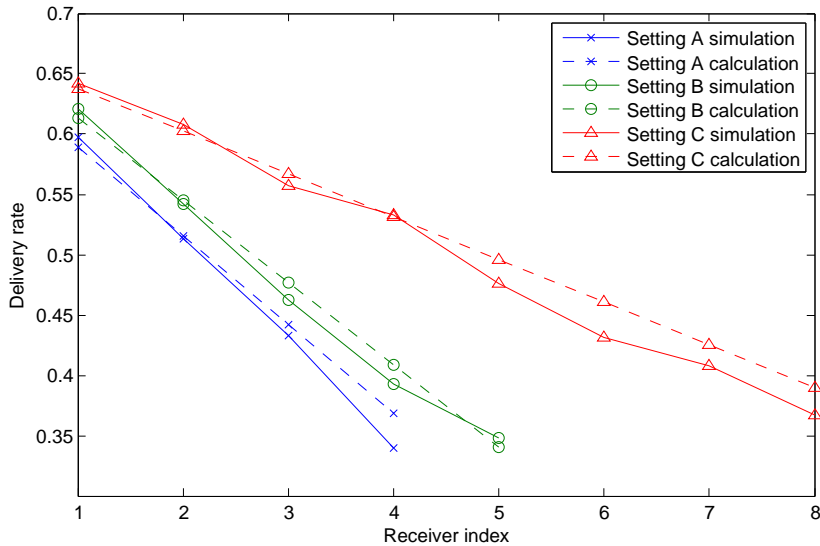


Fig. 9. Calculated and simulated delivery rates under the settings of Table VII, using the β values derived in Algorithm 6.

In Fig. 8 we can observe the delivery rate estimates after each iteration r in Algorithm 6. At each iteration r , the theoretical delivery rates are those that the sender would enforce if it only took into consideration the delivery ratios of the first r receivers. At iteration $r = 1$, R_{X_1} achieves the best possible performance with delivery rate $R_1 = \mu_1$, however the other receivers' delivery delays are significantly worse. At each subsequent value of r , the delivery rates of receivers $R_{X_1}, \dots, R_{X_{r-1}}$ are scaled down to allow R_{X_r} to match the delivery ratios of receivers $R_{X_1}, \dots, R_{X_{r-1}}$.

It is interesting to note that, after iteration r , there is improvement to the delivery rates of *all* mode r receivers (that is receivers R_{X_r} to R_{X_R}), not simply R_{X_r} . This occurs because in higher modes, the transmission $s(t)$ will be coded from a smaller set of receivers, whose next needed packets are more likely to have been seen. This improves the likelihood that a receiver will obtain a knowledge differential.

In Fig. 9 we compare the calculated and simulated delivery rates under the values of β determined from Algorithm 6. Similar to Fig. 7, we observe that for low receiver indices, estimates match well with calculated values, while some loss of accuracy is observed for higher receiver indices.

IX. CONCLUSION

In this paper we have demonstrated that, by relaxing the complete information acquisition requirement, it is possible to exceed the min-cut capacity of a broadcast packet erasure network. By implementing a deterministic coding scheme, we were

able to achieve multi-rate delivery of the same set of ordered information to a set of heterogeneous receivers, allowing the sender's transmission rate to exceed the channel rate of a number of receivers.

Despite the complexity associated with the analysis of deterministic coding schemes, through the use of some approximations we were able to obtain tractable estimates of the receivers' delivery rates. These estimates were shown to be reasonably accurate, although some loss of accuracy was observed with a larger number of receivers. Based on our approximate analysis, we introduced a fairness algorithm that determined how the sender should allocate its resources to achieve fair delivery rate performance for the receivers.

While the rate analysis we have presented in this paper lends insight into the process of multi-rate delivery, it is specific to the network coding scheme we have used. It remains unclear whether higher delivery rates could in fact be achieved using other deterministic transmission schemes. Future work should include the characterisation of the capacity of a broadcast network, given the relaxation of the complete information acquisition requirement.

REFERENCES

- [1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2002.
- [2] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 789–804, 2006.
- [3] D. Lun, M. Médard, and R. Koetter, "Network coding for efficient wireless unicast," in *Proc. International Zurich Seminar on Communications (IZS)*, 2006, pp. 74–77.
- [4] N. Shulman, "Communication over an unknown channel via common broadcasting," Ph.D. dissertation, Tel Aviv University, 2003.
- [5] S. Sanghavi, "Intermediate performance of rateless codes," in *Information Theory Workshop, 2007. ITW '07. IEEE*, 2007, pp. 478–482.
- [6] A. Beimel, S. Dolev, and N. Singer, "Rt oblivious erasure correcting," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1321–1332, 2007.
- [7] E. Martinian, "Dynamic information and constraints in source and channel coding," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.
- [8] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [9] J. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proc. Workshop on Network Coding, Theory, and Applications (NETCOD)*, 2009, pp. 1–6.
- [10] K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2008, pp. 1651–1655.
- [11] B. Swapna, A. Eryilmaz, and N. Shroff, "Throughput-delay analysis of random linear network coding for wireless broadcasting," in *Proc. International Symposium on Network Coding (NETCOD)*, June 2010, pp. 1–6.
- [12] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5511–5524, December 2008.
- [13] D. Lucani, M. Stojanovic, and M. Médard, "Random linear network coding for time division duplexing: When to stop talking and start listening," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, vol. 9, 2009, pp. 1800–1808.
- [14] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, vol. 4, 2005.
- [15] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [16] J. Park, M. Gerla, D. Lun, Y. Yi, and M. Médard, "Codecast: a network-coding-based ad hoc multicast protocol," *IEEE Wireless Commun. Mag.*, vol. 13, no. 5, pp. 76–81, 2006.
- [17] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Annual Allerton Conference on Communication, Control and Computing*, Oct. 2003.
- [18] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP Journal on Wireless Communications and Networking*, pp. 1–14, April 2010.
- [19] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. IEEE International Conference on Communications (ICC)*, May 2010, pp. 1–5.
- [20] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Workshop on Network Coding, Theory and Applications (NETCOD)*, June 2009, pp. 80–85.
- [21] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, May 2011.
- [22] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, IEEE, 2010, pp. 1–5.
- [23] W. Zeng, C. Ng, and M. Médard, "Joint coding and scheduling optimization in wireless systems with varying delay sensitivities," in *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012, pp. 416–424.
- [24] M. Durvy, C. Fragouli, and P. Thiran, "Towards reliable broadcasting using ACKs," in *IEEE International Symposium on Information Theory (ISIT)*, 2007, pp. 1156–1160.
- [25] J. K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network coding meets TCP: Theory and implementation," *Proc. IEEE*, vol. 99, no. 3, pp. 490–512, 2011.
- [26] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, April 2009, pp. 208–216.
- [27] A. Fu, P. Sadeghi, and M. Médard, "Dynamic rate adaptation for improved throughput and delay in wireless network coded broadcast," *IEEE/ACM Trans. Netw.*, to appear. [Online]. Available: <http://arxiv.org/abs/1208.3806>
- [28] F. Wu, C. Hua, H. Shan, and A. Huang, "Reliable network coding for minimizing decoding delay and feedback overhead in wireless broadcasting," in *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2012, pp. 796–801.
- [29] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, 2008.
- [30] N. Sundaram, P. Ramanathan, and S. Banerjee, "Multirate media streaming using network coding," in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, September 2005.
- [31] T. Cover, "Broadcast channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, 1972.
- [32] C. Fragouli, J. Le Boudec, and J. Widmer, "Network coding: An instant primer," *Computer Communication Review*, vol. 36, no. 1, p. 63, 2006.
- [33] M. Artin, *Algebra*, 1st ed. Prentice-Hall, 1991.
- [34] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.